

Implementasi AES-128 pada Sistem Pengelolaan Dokumen Notaris Berbasis Web

Panji Siwinindya Putra¹, Indah Puji Astuti², Adi Fajaryanto Cobantoro³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo
¹panjiputra105@gmail.com, ²indahpujiastuti@umpo.ac.id, ³adifajaryanto@umpo.ac.id

Abstrak

Pemanfaatan dokumen digital di lingkungan kantor notaris memberikan kemudahan dalam penyimpanan dan pengelolaan arsip secara sistematis. Di sisi lain, penggunaan dokumen elektronik juga menghadirkan risiko keamanan berupa akses oleh pihak yang tidak berwenang maupun potensi kebocoran informasi. Untuk meningkatkan perlindungan data digital, penelitian ini menerapkan algoritma *Advanced Encryption Standard* (AES-128) pada sistem pengelolaan dokumen notaris berbasis *web*. Metode pengembangan sistem yang digunakan adalah *Waterfall* yang meliputi analisis, perancangan, implementasi, dan pengujian. Sistem dikembangkan menggunakan PHP dan MySQL dengan mekanisme enkripsi otomatis saat dokumen diunggah serta dekripsi saat dokumen diakses melalui fitur *preview* dan *download*. Pendekatan tersebut mengintegrasikan proses enkripsi dan dekripsi ke dalam alur pengelolaan dokumen sehingga pengguna tidak perlu melakukan pengamanan secara manual. Pengujian dilakukan menggunakan *white box testing*, pengujian performa, dan validasi menggunakan *Devglan AES Encryption Tool*. Hasil pengujian menunjukkan bahwa seluruh fungsi enkripsi dan dekripsi berjalan sesuai dengan hasil yang diharapkan. Dokumen berukuran 88,79 KB membutuhkan waktu enkripsi sebesar 0,000182152 detik, sedangkan dokumen berukuran 1971,96 KB membutuhkan waktu enkripsi sebesar 0,00398803 detik. Hasil tersebut menunjukkan bahwa ukuran *file* memengaruhi waktu proses enkripsi dan dekripsi. Meskipun demikian, waktu pemrosesan yang dihasilkan tetap singkat sehingga tidak mengganggu kinerja sistem maupun akses dokumen oleh pengguna. Selain itu, hasil validasi menunjukkan bahwa implementasi AES-128 menghasilkan keluaran yang sesuai dengan *Devglan AES Encryption Tool*. Berdasarkan hasil penelitian, penerapan AES-128 mampu meningkatkan keamanan dokumen digital melalui mekanisme enkripsi dan dekripsi terintegrasi tanpa mengurangi kinerja sistem sehingga layak diterapkan pada sistem pengelolaan dokumen notaris berbasis *web*.

Kata kunci: algoritma AES-128, dokumen digital, keamanan dokumen

1. Pendahuluan

Pemanfaatan dokumen elektronik telah menjadi bagian penting dalam berbagai aktivitas administrasi, termasuk pada lingkungan kerja notaris dan PPAT. Sebagai pejabat umum yang berwenang membuat akta autentik, notaris memiliki tanggung jawab dalam mengelola dan menjaga dokumen yang berkaitan dengan pelayanan hukum kepada masyarakat (Abdillah & Sahindra, 2022). Berkas seperti KTP, Kartu Keluarga, sertifikat, dan dokumen pendukung lainnya umumnya disimpan dalam bentuk digital karena lebih mudah dikelola dibandingkan arsip fisik (Prakoso et al., 2023; Susena et al., 2023).

Meskipun memberikan kemudahan, penggunaan dokumen digital masih menghadapi berbagai tantangan yang berkaitan dengan keamanan informasi. Dokumen yang tersimpan dalam sistem memungkinkan terjadinya akses oleh pihak yang tidak berhak apabila tidak dilengkapi dengan mekanisme perlindungan yang memadai (Cobantoro, Islami, et al., 2025). Risiko tersebut menjadi penting karena sebagian besar dokumen yang dikelola oleh kantor

notaris berisi data pribadi dan dokumen yang memiliki nilai hukum (Novita et al., 2025). Berbagai kasus kebocoran dan penyalahgunaan data menunjukkan bahwa perlindungan terhadap dokumen digital menjadi kebutuhan yang semakin penting dalam pengelolaan sistem informasi (Nuranisa & Lukitarsi, 2024; Yuwana et al., 2025).

Perlindungan terhadap data digital dapat dilakukan melalui penerapan teknik kriptografi sebagai mekanisme pengamanan informasi. Melalui mekanisme ini, data diubah ke dalam bentuk *ciphertext* sehingga isi informasi tidak dapat dibaca secara langsung. Di antara berbagai algoritma kriptografi yang tersedia, *Advanced Encryption Standard* (AES) termasuk algoritma yang banyak digunakan karena memiliki tingkat keamanan yang baik dan proses komputasi yang relatif cepat (Alenezi et al., 2024). AES merupakan algoritma kriptografi simetris yang menggunakan kunci yang sama pada proses enkripsi dan dekripsi (Hutasuhut et al., 2023). Pada penelitian ini digunakan AES-128 yang memiliki panjang kunci 128 bit dan banyak diterapkan pada berbagai sistem pengamanan data digital (Nugraha & Arifin, 2022).

Algoritma AES-128 telah banyak diterapkan dalam penelitian sebelumnya untuk menjaga kerahasiaan berkas maupun data berbentuk teks. (Pradana, 2022) menerapkan AES untuk pengamanan *repository* dokumen digital pada lingkungan pengadilan militer. (Tarigan et al., 2024) menerapkan AES-128 untuk mengenkripsikan berkas bantuan sosial berbasis *web*, sedangkan (Tania et al., 2024) mengimplementasikan AES pada sistem keamanan data digital di kantor desa. (Ramadhani et al., 2024) juga menerapkan AES-128 untuk pengamanan *database* sistem registrasi pasien, sementara (Djami et al., 2025) menunjukkan bahwa AES-128 dapat digunakan untuk mengamankan data teks pada sistem keamanan data digital. Meskipun demikian, penerapan algoritma AES-128 pada penelitian-penelitian tersebut masih terbatas pada objek tertentu, seperti *repository* dokumen, berkas bantuan sosial, *database* pasien, dan data teks. Berdasarkan kajian terhadap penelitian terdahulu, implementasi algoritma AES-128 pada sistem pengelolaan dokumen notaris berbasis *web* masih jarang dibahas, terutama yang mengintegrasikan proses enkripsi dan dekripsi secara langsung ke dalam alur pengelolaan dokumen. Selain itu, pada beberapa penelitian proses enkripsi dan dekripsi masih dilakukan secara terpisah sehingga pengguna masih perlu melakukan pengamanan dokumen secara manual. Berdasarkan kondisi tersebut, masih terdapat peluang untuk mengembangkan sistem pengelolaan dokumen notaris berbasis *web* yang mengintegrasikan algoritma AES-128 secara langsung ke dalam proses pengelolaan dokumen.

Untuk menjawab kebutuhan tersebut, penelitian ini mengimplementasikan algoritma AES-128 pada sistem pengelolaan dokumen notaris berbasis *web*. Setiap dokumen yang diunggah akan dienkripsi secara otomatis sebelum disimpan ke *server*, sedangkan proses dekripsi dilakukan ketika dokumen diakses melalui fitur *preview* maupun *download*. Dengan mekanisme tersebut, dokumen tetap tersimpan dalam bentuk terenkripsi dan hanya dapat ditampilkan kembali setelah melalui proses dekripsi. Pendekatan ini memungkinkan mekanisme pengamanan diterapkan secara otomatis pada proses pengelolaan dokumen tanpa menambah tahapan kerja bagi pengguna. Kontribusi penelitian ini terletak pada penerapan algoritma AES-128 yang terintegrasi ke dalam proses unggah, penyimpanan, *preview*, dan *download* dokumen sehingga mekanisme pengamanan dapat diterapkan secara otomatis pada sistem pengelolaan dokumen notaris berbasis *web*.

Pengujian dilakukan menggunakan *white box testing*, pengujian performa, serta validasi dengan *Devglan AES Encryption Tool* untuk memastikan bahwa proses enkripsi dan dekripsi pada sistem berjalan sesuai dengan rancangan. Selain menguji fungsionalitas sistem, pengujian performa digunakan untuk mengetahui pengaruh ukuran dokumen terhadap waktu pemrosesan, sedangkan validasi dilakukan

dengan membandingkan hasil implementasi AES-128 pada sistem dengan *Devglan AES Encryption Tool*. Hasil pengujian tersebut diharapkan dapat menunjukkan bahwa mekanisme pengamanan yang diterapkan mampu meningkatkan keamanan dokumen digital tanpa mengganggu proses pengelolaan dokumen di lingkungan kantor notaris.

2. Metode

2.1 Tahapan Penelitian

Proses penelitian dilaksanakan melalui serangkaian tahapan yang meliputi identifikasi masalah, studi literatur, pengumpulan data, serta pengembangan sistem menggunakan metode *Waterfall* yang terdiri atas tahap analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Tahapan tersebut dilakukan untuk mendukung proses pengembangan sistem pengelolaan dokumen berbasis *web* menggunakan algoritma AES-128 (Supriatiningsih, 2023).

Tahap awal penelitian dilakukan dengan mengamati proses pengelolaan dokumen yang berlangsung pada kantor notaris untuk mengidentifikasi permasalahan yang ada. Dari hasil observasi diketahui bahwa dokumen digital telah digunakan dalam proses administrasi, namun belum terdapat mekanisme pengamanan data yang diterapkan secara langsung pada sistem pengelolaan dokumen.

Selanjutnya dilakukan studi literatur melalui penelusuran berbagai referensi yang berkaitan dengan AES-128 dan teknologi pendukung lainnya. Studi literatur memberikan landasan teoritis dalam proses pengembangan sistem serta implementasi algoritma AES-128 (Alenezi et al., 2024; Nugraha & Arifin, 2022).

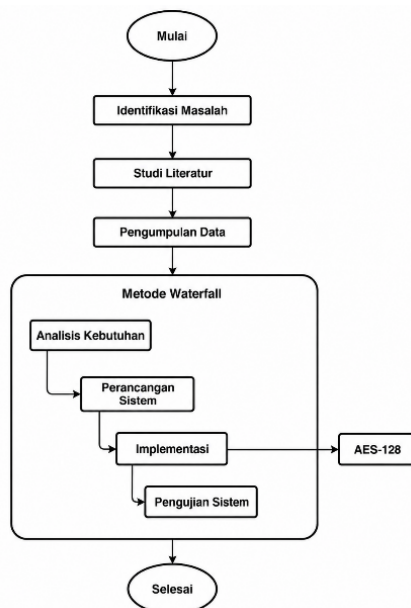
Selain studi literatur, dilakukan pula pengumpulan data melalui observasi dan dokumentasi untuk memperoleh informasi mengenai proses pengelolaan dokumen, kebutuhan pengguna, serta kebutuhan keamanan dokumen digital. Data yang diperoleh digunakan sebagai dasar dalam pengembangan sistem. Pengembangan sistem dilakukan menggunakan metode *Waterfall* yang terdiri atas tahap analisis kebutuhan, perancangan sistem, implementasi, dan pengujian.

Berdasarkan data yang telah diperoleh, dilakukan analisis kebutuhan sistem yang meliputi kebutuhan pengguna, proses pengelolaan dokumen, serta kebutuhan keamanan data digital. Hasil analisis kemudian menjadi dasar dalam proses perancangan dan implementasi sistem menggunakan PHP dan MySQL.

Tahapan akhir penelitian berupa validasi sistem dilakukan melalui kegiatan pengujian yang bertujuan menilai kesesuaian kinerja setiap fungsi dengan tujuan pengembangan yang telah ditetapkan. Proses tersebut dilaksanakan dengan menggunakan *white*

box testing untuk memeriksa fungsi sistem (Agustin et al., 2023; Cobantoro, Wicahyono, et al., 2025), pengujian performa untuk mengetahui pengaruh ukuran *file* terhadap waktu proses enkripsi dan dekripsi, serta validasi hasil implementasi menggunakan *Devglan AES Encryption Tool*.

Berdasarkan tahapan yang telah diuraikan, penelitian dilaksanakan secara bertahap mulai dari identifikasi masalah, studi literatur, pengumpulan data, hingga pengembangan sistem menggunakan metode *Waterfall*. Rangkaian tahapan penelitian tersebut ditunjukkan pada Gambar 1.



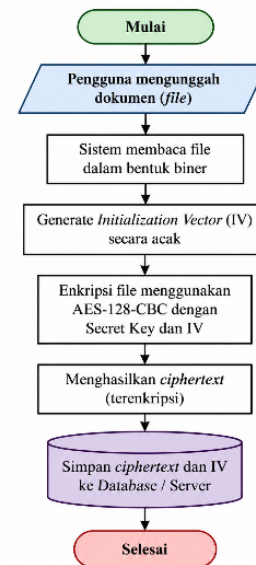
Gambar 1. Tahapan Penelitian

2.2 Alur Enkripsi dan Dekripsi Dokumen

Mekanisme enkripsi pada sistem dijalankan secara otomatis setiap kali pengguna melakukan unggah dokumen. *File* yang diunggah akan dibaca dalam bentuk *binary* kemudian diproses menggunakan AES-128 dengan mode CBC sebelum disimpan pada *server*. AES-128 dipilih karena memiliki tingkat keamanan yang baik dan performa yang efisien dalam proses pengamanan data digital (Alenezi et al., 2024; Nugraha & Arifin, 2022).

Sebelum data dienkripsi, sistem terlebih dahulu membangkitkan *Initialization Vector* (IV) yang digunakan pada mode CBC. Setelah proses enkripsi selesai, *file* akan disimpan dalam bentuk *ciphertext* sehingga isi dokumen tidak dapat dibaca secara langsung. Alur proses enkripsi dokumen pada sistem ditunjukkan pada Gambar 2.

Sedangkan proses dekripsi dijalankan ketika pengguna membuka dokumen menggunakan fasilitas *preview* maupun *download*. Sistem akan mengambil dokumen yang telah diamankan melalui mekanisme enkripsi dari *server*, kemudian melakukan proses dekripsi menggunakan *secret key* yang sama sehingga



Gambar 2. Flowchart Enkripsi Dokumen

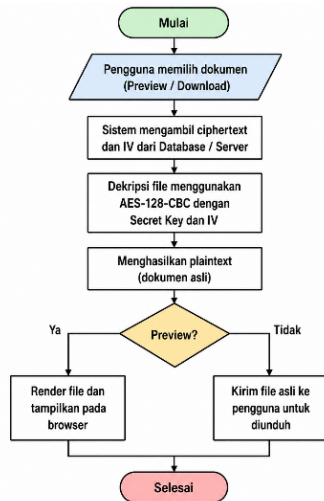
dokumen dapat ditampilkan kembali dalam bentuk aslinya. Pada penelitian ini, proses *preview* dan *download* menggunakan mekanisme dekripsi yang sama, namun proses *preview* memerlukan proses *rendering* dokumen pada *browser* sebelum *file* dapat ditampilkan kepada pengguna. Alur proses dekripsi dokumen pada sistem ditunjukkan pada Gambar 3.

2.3 Implementasi AES-128 pada Sistem

Implementasi keamanan dokumen pada sistem dilakukan menggunakan bahasa pemrograman PHP versi 8.2.12 dengan basis data MariaDB versi 10.4.32 yang dijalankan pada *web server* Apache 2.4.58. Proses pengembangan dan pengujian sistem dilakukan menggunakan laptop Acer Aspire A514-52G dengan prosesor Intel® Core™ i5-10210U CPU @ 1.60 GHz dan RAM 8 GB. Implementasi algoritma AES-128 dilakukan menggunakan pustaka OpenSSL pada PHP dalam mode *Cipher Block Chaining* (CBC) dengan panjang kunci 128 bit.

Sistem memproses dokumen berformat PDF yang telah melalui proses validasi sebelum dilakukan enkripsi. Ketika pengguna melakukan unggah dokumen, sistem membaca isi *file* dalam bentuk data biner kemudian memprosesnya menggunakan algoritma AES-128 mode CBC. Setiap proses enkripsi menggunakan *Initialization Vector* (IV) sepanjang 16 *byte* yang dibangkitkan secara acak menggunakan fungsi *random_bytes()*. Data biodata berupa NIK, nama, dan alamat dienkripsi menjadi *ciphertext* dalam format *hexadecimal*, sedangkan isi dokumen dienkripsi menjadi *ciphertext* yang dikodekan menggunakan Base64 sebelum disimpan pada basis data bersama nilai *Initialization Vector* (IV) yang digunakan kembali pada proses dekripsi.

Saat dokumen diakses kembali, sistem melakukan proses dekripsi menggunakan *secret*



Gambar 3. Flowchart Dekripsi Dokumen

key yang sama dengan proses enkripsi sehingga dokumen dapat dikembalikan ke bentuk aslinya. Pada fitur *preview*, hasil dekripsi ditampilkan langsung melalui *browser* sehingga pengguna dapat melihat isi dokumen tanpa mengunduh *file*. Sementara itu, pada fitur *download*, dokumen yang telah didekripsi dikirim kembali kepada pengguna dalam bentuk *file* asli.

Dengan mekanisme tersebut, proses pengamanan data dapat berlangsung secara otomatis tanpa memerlukan langkah tambahan dari pengguna, sehingga keamanan dokumen tetap terjaga selama proses penyimpanan maupun akses dokumen.

2.4 Pengujian Sistem

Tahap pengujian dilakukan untuk mengevaluasi keberhasilan penerapan AES-128 pada sistem pengelolaan dokumen notaris. Pengujian difokuskan pada aspek fungsionalitas, performa, serta validitas hasil implementasi algoritma.

Evaluasi fungsional dilakukan menggunakan metode *white box testing* terhadap fungsi-fungsi utama yang terlibat dalam proses enkripsi dan dekripsi. Pengujian ini bertujuan memastikan bahwa setiap alur program berjalan sesuai dengan logika yang telah dirancang pada tahap implementasi.

Selain pengujian fungsional, dilakukan pula pengujian performa dengan mengukur waktu yang diperlukan sistem dalam proses enkripsi dan dekripsi menggunakan fungsi *microtime(true)* pada PHP. Waktu pemrosesan dihitung berdasarkan selisih waktu sebelum dan sesudah proses enkripsi maupun dekripsi dijalankan. Pengujian dilakukan menggunakan dokumen berformat PDF dengan ukuran *file* yang bervariasi untuk mengetahui pengaruh ukuran *file* terhadap durasi proses enkripsi dan dekripsi algoritma AES-128. Pada fitur *preview* dan *download*, waktu dekripsi diukur secara terpisah sehingga performa masing-masing proses dapat dievaluasi.

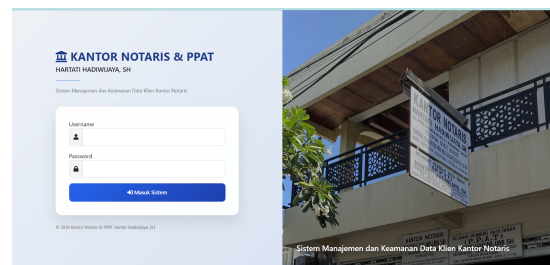
Validasi hasil implementasi dilakukan menggu-

nakan *Devglan AES Encryption Tool* sebagai alat pembanding. Proses ini dilakukan dengan membandingkan hasil enkripsi dan dekripsi yang dihasilkan sistem terhadap hasil yang diperoleh dari *Devglan AES Encryption Tool* menggunakan parameter AES-128 yang sama.

3. Hasil dan Pembahasan

3.1 Implementasi Sistem

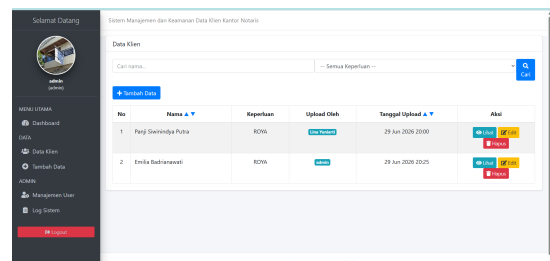
Hasil penelitian berupa sistem pengelolaan dokumen notaris berbasis *web* yang digunakan untuk mengelola data klien dan dokumen pendukung administrasi. Sistem dibangun menggunakan PHP dan MySQL serta dilengkapi dengan mekanisme pengamanan dokumen menggunakan algoritma AES-128.



Gambar 4. Halaman Login

Antarmuka autentikasi pengguna diperlihatkan pada Gambar 4 yang berfungsi sebagai pintu masuk ke dalam sistem. Pada halaman ini pengguna harus memasukkan *username* dan *password* yang telah terdaftar sebelum dapat mengakses fitur yang tersedia.

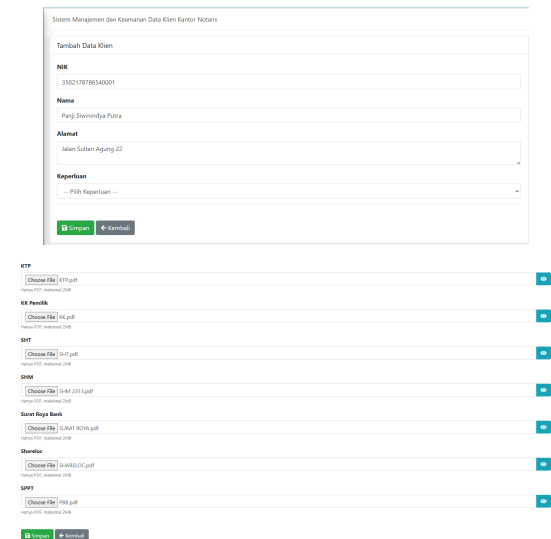
Setelah berhasil *login*, pengguna dapat mengakses halaman data klien seperti yang ditunjukkan pada Gambar 5. Halaman ini digunakan untuk menyimpan dan mengelola informasi klien yang berkaitan dengan dokumen yang akan diproses pada sistem. Data yang tersimpan pada halaman ini digunakan sebagai acuan dalam proses unggah dan pengelolaan dokumen sehingga setiap dokumen dapat terhubung dengan identitas klien yang sesuai.



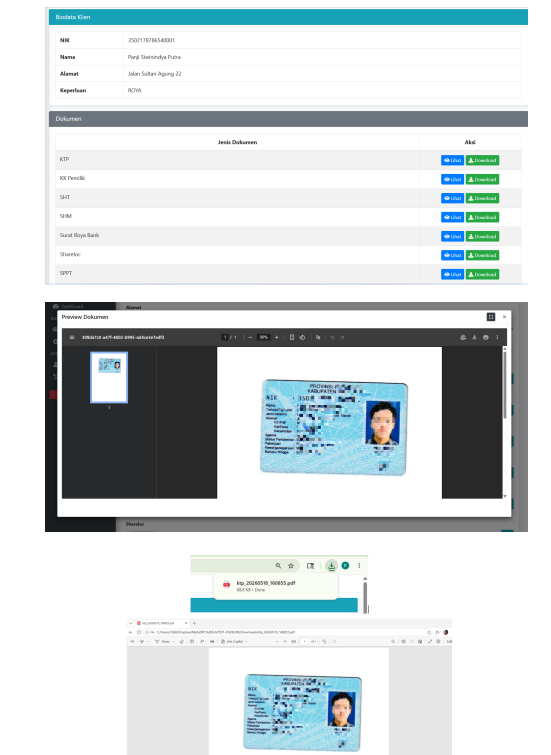
Gambar 5. Halaman Data Klien

Halaman unggah dokumen yang digunakan untuk memasukkan berkas ke dalam sistem dapat dilihat pada Gambar 6. Melalui halaman ini pengguna dapat mengunggah dokumen seperti KTP, KK, sertifikat, maupun dokumen pendukung lainnya. Saat proses unggah dilakukan, sistem secara otomatis mengenkripsi *file* menggunakan AES-128 sebelum

disimpan pada *server*. Hasil enkripsi berupa *ciphertext* disimpan pada media penyimpanan sehingga dokumen tidak dapat diakses secara langsung tanpa melalui proses dekripsi. Penerapan mekanisme ini bertujuan untuk menjaga kerahasiaan dokumen selama proses penyimpanan pada sistem.



Gambar 6. Halaman *Upload* Dokumen



Gambar 7. Halaman *Preview* dan *Download* Dokumen

Dokumen yang telah tersimpan dalam sistem dapat diakses kembali melalui fitur *preview* maupun *download*. Sebelum dokumen ditampilkan atau diunduh, sistem terlebih dahulu melakukan proses dekripsi menggunakan *secret key* yang sesuai sehingga dokumen dapat dikembalikan ke bentuk aslinya. Halaman *preview* dan *download* dokumen

ditunjukkan pada Gambar 7. Kedua fitur tersebut dirancang untuk memenuhi kebutuhan akses dokumen tanpa mengubah mekanisme penyimpanan yang tetap menggunakan data terenkripsi. Dengan demikian, kerahasiaan dokumen tetap terjaga selama proses penyimpanan, sedangkan dokumen asli hanya ditampilkan ketika diakses oleh pengguna yang memiliki hak akses.

Melalui halaman tersebut, pengguna dapat melihat isi dokumen melalui fitur *preview* ataupun memperoleh dokumen asli melalui fitur *download*. Kedua proses tersebut menggunakan mekanisme dekripsi yang sama sehingga dokumen tetap tersimpan dalam keadaan terenkripsi pada *server* dan hanya dikembalikan ke bentuk aslinya ketika diakses oleh pengguna yang memiliki hak akses.

Penerapan mekanisme enkripsi secara otomatis pada saat proses unggah memungkinkan dokumen tetap tersimpan dalam bentuk terenkripsi selama berada pada media penyimpanan. Pendekatan tersebut sejalan dengan penelitian (Tarigan et al., 2024) yang juga menerapkan AES-128 untuk mengamankan dokumen digital melalui proses enkripsi sebelum penyimpanan. Perbedaannya, pada penelitian ini mekanisme enkripsi dan dekripsi diintegrasikan secara otomatis ke dalam alur pengelolaan dokumen secara otomatis ke dalam alur pengelolaan dokumen notaris berbasis *web*, sehingga pengguna tidak perlu melakukan proses pengamanan dokumen secara manual.

3.2 Pengujian *White Box*

Metode *white box testing* diterapkan untuk memastikan bahwa fungsi-fungsi utama yang berkaitan dengan proses enkripsi dan dekripsi berjalan sesuai dengan rancangan. Pengujian difokuskan pada fungsi *encryptText()*, *encryptFile()*, *decryptText()*, dan *decryptFile()* untuk memastikan bahwa setiap fungsi dapat menghasilkan keluaran sesuai dengan yang dirancang.

Tabel 1. Hasil Pengujian *White Box*

Method	Input	Hasil Pengujian	Status
<i>encryptText()</i>	Panji Siwinindya Putra	Data berhasil dienkripsi menjadi <i>ciphertext</i>	Valid
<i>encryptFile()</i>	File KTP	File berhasil dienkripsi menjadi <i>ciphertext</i>	Valid
<i>decryptText()</i>	<i>Ciphertext</i>	Data berhasil dikembalikan ke bentuk <i>plaintext</i>	Valid
<i>decryptFile()</i> (Preview)	File terenkripsi	File berhasil ditampilkan dalam bentuk asli	Valid
<i>decryptFile()</i> (Download)	File terenkripsi	File berhasil diunduh dalam bentuk asli	Valid

Tabel 1 menunjukkan hasil pengujian terhadap

fungsi-fungsi yang digunakan dalam implementasi AES-128 pada sistem.

Salah satu hasil pengujian ditunjukkan pada Gambar 8. Pengujian dilakukan menggunakan fungsi *encryptText()* dengan data masukan berupa nama klien “Panji Siwinindya Putra”. Hasil pengujian menunjukkan bahwa data berhasil diubah menjadi *ciphertext* sebelum disimpan ke dalam basis data. Kondisi tersebut menunjukkan bahwa proses enkripsi berjalan sesuai dengan rancangan sehingga informasi tidak dapat dibaca secara langsung ketika tersimpan pada sistem.

```
nama
94B04CB0BFE3AB5DAF052F053DE054D3875DEF58A6F272B8D7...
iv_nama
F80757967243E18CE26BDE9ED4279928
```

Gambar 8. Hasil Pengujian *encryptText()*

Hasil pengujian yang disajikan pada Tabel 1 menunjukkan bahwa seluruh fungsi yang diuji mampu menjalankan proses sesuai dengan kebutuhan sistem yang telah ditetapkan. Fungsi enkripsi berhasil mengubah data dan dokumen menjadi *ciphertext*, sedangkan fungsi dekripsi mampu mengembalikan data ke bentuk semula tanpa mengalami perubahan isi. Seluruh skenario pengujian menghasilkan keluaran sesuai dengan hasil yang diharapkan (*expected result*), sehingga implementasi algoritma AES-128 pada sistem dinyatakan *valid*.

Hasil tersebut sejalan dengan penelitian (Cobantoro, Wicahyono, et al., 2025) yang juga menerapkan *white box testing* pada implementasi algoritma kriptografi dan memperoleh hasil pengujian yang menunjukkan bahwa fungsi-fungsi sistem berjalan sesuai dengan logika program yang dirancang. Selain itu, hasil penelitian ini juga konsisten dengan penelitian (Agustin et al., 2023) yang menunjukkan bahwa *white box testing* mampu memverifikasi fungsi-fungsi program melalui pengujian terhadap *class* dan *method* sehingga setiap fungsi dapat dijalankan sesuai dengan rancangan sistem. Kesamaan hasil tersebut menunjukkan bahwa *white box testing* efektif digunakan untuk memverifikasi logika internal program serta memastikan implementasi algoritma AES-128 pada sistem telah berjalan sesuai dengan rancangan.

3.3 Pengujian Performa AES-128

Evaluasi performa dilakukan untuk menganalisis hubungan antara ukuran dokumen dan waktu pemrosesan yang diperlukan pada proses enkripsi maupun dekripsi. Pengujian menggunakan beberapa dokumen dengan ukuran yang berbeda, mulai dari dokumen berukuran kecil hingga dokumen berukuran besar. Parameter yang diamati meliputi ukuran dokumen, waktu enkripsi, waktu dekripsi *preview*, dan waktu dekripsi *download*.

Data pada Tabel 2 menunjukkan bahwa ukuran dokumen memengaruhi durasi proses enkripsi dan

dekripsi. Dokumen dengan ukuran yang lebih besar membutuhkan waktu pemrosesan yang lebih lama dibandingkan dokumen berukuran kecil. Hal tersebut menunjukkan bahwa jumlah data yang diproses oleh algoritma AES-128 berpengaruh terhadap waktu komputasi yang diperlukan. Kondisi tersebut terjadi karena AES-128 merupakan algoritma kriptografi berbasis blok (*block cipher*) yang memproses data dalam blok berukuran 128 bit (16 *byte*). Semakin besar ukuran dokumen, semakin banyak blok data yang harus diproses selama enkripsi maupun dekripsi sehingga waktu komputasi yang dibutuhkan juga meningkat. Meskipun demikian, peningkatan waktu tersebut masih bersifat proporsional terhadap ukuran dokumen dan tidak menunjukkan penurunan performa yang signifikan.

Tabel 2. Hasil Pengujian Performa AES-128

Data	KB	Enk (s)	Prev (s)	Down (s)	Status
Ktp	88,79	0,000182152	0,000076055	0,000083923	Berhasil
Kk	288,58	0,000430107	0,000440836	0,000207901	Berhasil
Sht	1280,87	0,002527	0,00145912	0,00123191	Berhasil
Shm	1971,96	0,00398803	0,00196099	0,00254583	Berhasil
Surat Roya	172,83	0,000271082	0,000130892	0,000133991	Berhasil
Shareloc	167,49	0,000266075	0,000201940	0,000189066	Berhasil
Sppt	267,76	0,000391006	0,000483036	0,00014782	Berhasil

Keterangan: KB = Ukuran dokumen (*Kilobyte*); Enk = Waktu enkripsi (detik); Prev = Waktu dekripsi *preview* (detik); Down = Waktu dekripsi *download* (detik)

Pada proses enkripsi, dokumen KTP dengan ukuran 88,79 KB membutuhkan waktu enkripsi sebesar 0,000182152 detik. Sementara itu, dokumen SHM dengan ukuran 1971,96 KB membutuhkan waktu enkripsi sebesar 0,00398803 detik. Temuan tersebut mengindikasikan bahwa waktu komputasi cenderung meningkat ketika ukuran dokumen yang diproses semakin besar.

Hasil tersebut sejalan dengan penelitian (Tarigan et al., 2024) yang menunjukkan bahwa peningkatan ukuran *file* berpengaruh terhadap bertambahnya waktu pemrosesan algoritma AES-128. Hasil tersebut juga mendukung penelitian (Alenezi et al., 2024) yang menyatakan bahwa AES merupakan algoritma kriptografi simetris yang memiliki efisiensi komputasi tinggi sehingga tetap mampu memberikan performa yang baik meskipun digunakan pada ukuran data yang berbeda. Dengan demikian, hasil pengujian pada penelitian ini menunjukkan bahwa penerapan AES-128 tidak hanya mampu meningkatkan keamanan dokumen, tetapi juga tetap mempertahankan performa sistem pada proses pengelolaan dokumen berbasis *web*.

Pengujian juga dilakukan pada proses dekripsi melalui fitur *preview* dan *download*. Secara umum, waktu dekripsi yang dihasilkan masih berada pada rentang yang relatif kecil. Perbedaan waktu antara kedua fitur tersebut dipengaruhi oleh mekanisme

yang dijalankan sistem setelah proses dekripsi selesai. Pada fitur *preview*, dokumen yang telah didekripsi diproses kembali untuk ditampilkan secara langsung (*inline*) pada *browser* sehingga terdapat tambahan proses penyajian dokumen. Sebaliknya, pada fitur *download*, dokumen langsung dikirim kepada pengguna sebagai berkas (*attachment*) tanpa melalui proses penampilan isi dokumen di *browser*. Perbedaan alur tersebut menyebabkan waktu dekripsi pada kedua fitur tidak selalu sama meskipun menggunakan mekanisme dekripsi AES-128 yang identik.

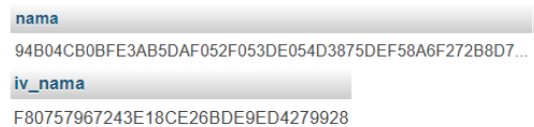
Berdasarkan Tabel 2, seluruh dokumen yang diuji memperoleh status berhasil pada proses enkripsi maupun dekripsi. Waktu enkripsi berada pada rentang 0,000182152 detik hingga 0,00398803 detik, sedangkan waktu dekripsi *preview* berada pada rentang 0,0000760555 detik hingga 0,00196099 detik. Adapun waktu dekripsi *download* berada pada rentang 0,0000839233 detik hingga 0,00254583 detik. Temuan tersebut menunjukkan bahwa mekanisme enkripsi dan dekripsi yang diterapkan masih berada pada rentang waktu yang relatif singkat sehingga tidak memberikan pengaruh yang berarti terhadap kenyamanan pengguna ketika mengakses dokumen melalui sistem. Hasil pengujian menunjukkan bahwa waktu enkripsi maupun dekripsi masih berada pada rentang yang sangat singkat, yaitu kurang dari 0,005 detik untuk seluruh dokumen yang diuji. Kondisi tersebut menunjukkan bahwa penerapan AES-128 masih layak digunakan pada sistem pengelolaan dokumen notaris berbasis *web* karena mekanisme pengamanan dokumen dapat diterapkan tanpa mengurangi responsivitas sistem secara signifikan ketika pengguna melakukan unggah, *preview*, maupun *download* dokumen.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan. Pengujian dilakukan menggunakan satu spesifikasi perangkat keras dengan jumlah dokumen uji yang terbatas serta hanya menggunakan dokumen berformat PDF. Selain itu, pengujian belum dilakukan pada kondisi banyak pengguna yang mengakses sistem secara bersamaan (*concurrent users*), sehingga performa sistem pada beban kerja yang lebih tinggi belum dapat dievaluasi secara menyeluruh. Oleh karena itu, penelitian selanjutnya dapat mengembangkan pengujian dengan variasi format dokumen, spesifikasi perangkat, serta skenario akses secara bersamaan untuk memperoleh gambaran performa sistem yang lebih komprehensif.

3.4 Validasi Menggunakan Devglan Tool

Proses validasi dilakukan untuk memverifikasi kesesuaian hasil implementasi AES-128 yang diterapkan pada sistem dengan implementasi standar yang tersedia pada *Devglan AES Encryption Tool*. Pengujian menggunakan data nama klien “Panji Siwinindya Putra” sebagai data masukan dengan konfigurasi algoritma dan parameter yang sama pada kedua *platform*.

Hasil enkripsi yang diperoleh dari sistem ditampilkan pada Gambar 9, sedangkan hasil enkripsi menggunakan *Devglan AES Encryption Tool* ditunjukkan pada Gambar 10. Perbandingan kedua hasil tersebut memperlihatkan bahwa *ciphertext* yang dihasilkan memiliki nilai yang sama. Hasil yang diperoleh mengindikasikan bahwa implementasi AES-128 pada sistem telah berjalan sesuai dengan mekanisme algoritma yang digunakan. Selain itu, tidak ditemukan perbedaan hasil yang mengindikasikan adanya kesalahan pada proses pembangkitan *ciphertext*. Oleh karena itu, mekanisme enkripsi yang diterapkan dapat menghasilkan keluaran yang konsisten dengan *Devglan AES Encryption Tool*. Kesesuaian tersebut menunjukkan bahwa proses enkripsi pada sistem telah menerapkan parameter AES-128 secara tepat sehingga menghasilkan *ciphertext* yang identik dengan implementasi standar.



```
nama
94B04CB0BFE3AB5DAF052F053DE054D3875DEF58A6F272B8D7...
iv_nama
F80757967243E18CE26BDE9ED4279928
```

Gambar 9. Hasil Enkripsi pada Database Sistem

Berdasarkan perbandingan tersebut, tidak ditemukan perbedaan keluaran antara sistem dan *Devglan AES Encryption Tool*. Selanjutnya, dilakukan pengujian pada proses dekripsi untuk memastikan bahwa data dapat dikembalikan ke bentuk semula. Hasil dekripsi menggunakan *Devglan AES Encryption Tool* ditunjukkan pada Gambar 11. Analisis pengujian menunjukkan bahwa *ciphertext* berhasil didekripsi menjadi *plaintext* tanpa mengalami perubahan isi. Kesesuaian keluaran yang diperoleh dari sistem dan *Devglan AES Encryption Tool* menunjukkan bahwa proses dekripsi telah berjalan sesuai dengan mekanisme AES-128. Hasil tersebut memperkuat bahwa mekanisme dekripsi yang diterapkan mampu mengembalikan data ke bentuk aslinya secara tepat tanpa mengubah isi informasi. Dengan demikian, implementasi dekripsi pada sistem dapat menjaga integritas data selama proses pengembalian dokumen ke bentuk aslinya.

Berdasarkan hasil validasi tersebut, sistem berhasil menjalankan enkripsi dan dekripsi secara tepat sehingga dapat digunakan untuk mendukung pengamanan dokumen digital pada sistem pengelolaan dokumen notaris berbasis *web*. Kesesuaian *ciphertext* yang dihasilkan menunjukkan bahwa parameter AES-128, penggunaan *key*, dan *Initialization Vector* (IV) telah diterapkan dengan benar sehingga mekanisme enkripsi dan dekripsi dapat berjalan secara konsisten.

Hasil validasi tersebut sejalan dengan penelitian (Djami et al., 2025) yang menunjukkan bahwa implementasi AES-128 mampu menghasilkan proses enkripsi dan dekripsi yang konsisten ketika parameter algoritma diterapkan dengan benar. Selain itu, (Nugraha & Arifin, 2022) juga menjelaskan bahwa

The screenshot shows the 'AES Encryption' tool interface. It includes fields for 'Enter Plain Text to Encrypt' (containing 'Panji Siwinindya Putra'), 'Enter Secret Key' (containing 'PENGOLAHANDATAKU'), and 'Enter IV (Optional)' (containing 'F80757967243E18CE26BDE9ED4279928'). There are buttons for 'Generate' and 'Encrypt'. The 'Secret Key Format' is set to 'Plain Text', 'Select Cipher Mode of Encryption' is 'CBC', 'Select Padding' is 'PKCS5Padding', 'Key Size in Bits' is '128', and 'Output Text Format' is 'Hex'. The 'AES Encrypted Output' is displayed as a long hexadecimal string.

Gambar 10. Hasil Enkripsi Menggunakan Devglan AES Encryption Tool

The screenshot shows the 'AES Decryption' tool interface. It includes fields for 'Enter Secret Key used for Encryption' (containing 'PENGOLAHANDATAKU') and 'Enter IV Used During Encryption(Optional)' (containing 'F80757967243E18CE26BDE9ED4279928'). There are buttons for 'Decrypt' and 'Copy'. The 'Secret Key Format' is set to 'Plain Text', 'Select Cipher Mode of Decryption' is 'CBC', 'Select Padding' is 'PKCS5Padding', 'Key Size in Bits' is '128', and 'Output Text Format' is 'Plain-Text'. The 'AES Decrypted Output' is displayed as 'Panji Siwinindya Putra'.

Gambar 11. Hasil Dekripsi Menggunakan Devglan AES Encryption Tool

penggunaan AES-128 dengan konfigurasi algoritma yang tepat mampu menghasilkan proses enkripsi dan dekripsi yang *valid*. Temuan tersebut menunjukkan bahwa implementasi AES-128 pada sistem tidak hanya menghasilkan *ciphertext* yang sesuai dengan *Devglan AES Encryption Tool*, tetapi juga menunjukkan bahwa konfigurasi algoritma, penggunaan *key*, dan *Initialization Vector* (IV) telah diterapkan secara benar sehingga mekanisme enkripsi dan dekripsi dapat bekerja secara konsisten.

Selain keberhasilan proses enkripsi dan dekripsi, aspek pengelolaan kunci (*key management*) juga menjadi faktor penting dalam menjaga keamanan sistem. Pada implementasi penelitian ini digunakan satu *secret key* yang sama untuk proses enkripsi dan dekripsi. Pendekatan tersebut digunakan sebagai bagian dari implementasi sistem dalam penelitian ini. Namun, apabila *secret key* tersebut diketahui oleh pihak yang tidak berwenang, seluruh dokumen yang tersimpan berpotensi untuk didekripsi. Oleh karena itu, pada implementasi di lingkungan produksi diperlukan mekanisme *key management* yang lebih optimal, misalnya melalui penyimpanan kunci di luar kode program menggunakan *environment variables* atau layanan manajemen kunci (*Key Management System*) untuk meningkatkan keamanan pengelolaan dokumen.

4. Kesimpulan

Penelitian ini menghasilkan sistem pengelolaan dokumen notaris berbasis *web* yang menerapkan algoritma AES-128 sebagai mekanisme pengamanan dokumen digital. Proses enkripsi dilakukan secara otomatis saat dokumen diunggah, sedangkan dekripsi dilakukan ketika dokumen diakses melalui fitur *pre-view* maupun *download*. Hasil pengujian menunjukkan bahwa seluruh fungsi enkripsi dan dekripsi berjalan sesuai dengan rancangan sistem. Pengujian performa juga menunjukkan bahwa ukuran dokumen memengaruhi durasi proses enkripsi dan dekripsi, meskipun seluruh dokumen yang diuji tetap dapat diproses dalam waktu yang relatif singkat. Validasi menggunakan *Devglan AES Encryption Tool* menghasilkan *ciphertext* dan *plaintext* yang sesuai dengan keluaran sistem, sehingga implementasi AES-128 pada sistem terbukti berjalan sesuai dengan mekanisme algoritma yang digunakan. Berdasarkan hasil tersebut, AES-128 dapat digunakan sebagai solusi untuk meningkatkan keamanan dokumen digital pada lingkungan kantor notaris. Penelitian ini masih memiliki keterbatasan, yaitu pengujian hanya dilakukan menggunakan satu spesifikasi perangkat keras, jumlah dokumen uji yang terbatas, serta hanya menggunakan dokumen berformat PDF dan belum mengevaluasi performa sistem pada kondisi akses oleh banyak pengguna secara bersamaan (*concurrent users*). Oleh karena itu, penelitian selanjutnya dapat dikembangkan melalui pengujian pada berba-

gai format dokumen, variasi spesifikasi perangkat, serta skenario akses multi-pengguna untuk memperoleh evaluasi performa yang lebih komprehensif. Selain itu, sistem dapat dikembangkan dengan menerapkan mekanisme *key management* yang lebih aman serta membandingkan algoritma AES-128 dengan algoritma kriptografi lainnya untuk mengevaluasi aspek keamanan dan performa secara lebih menyeluruh.

Daftar Pustaka

- Abdillah, S., & Sahindra, R. (2022): *Spesialisitas notaris-PPAT sebagai profesi dan jabatan dalam sistem hukum Indonesia*, Jurnal Hukum dan Kenotariatan, 1(1), 1–10.
- Agustin, N. D., Cobantoro, A. F., Setyawan, M. B., & Nurfitri, K. (2023): *Penerapan algoritma linear search di aplikasi SecondHand*, Jurnal Ilmiah NERO, 8(2), 107–116.
- Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2024): *On the performance of AES algorithm variants*, International Journal of Information and Computer Security, 23(3), 322–337. <https://doi.org/10.1504/IJICS.2024.138494>
- Cobantoro, A. F., Islami, M. V. D. L., Munaji, Arifin, R., & Setyawan, M. B. (2025): *Implementation of RFID technology as an innovative solution for payment systems in supermarkets*, International Journal of Information System and Technology, 1(5), 242–251.
- Cobantoro, A. F., Wicahyono, F. Y., & Abdurrozzaq, I. (2025): *Implementasi algoritma RSA untuk keamanan data pasien menggunakan teknologi QR code*, Jurnal Ilmiah NERO, 10(1), 37–48.
- Djami, D., Ninu, I., Naitkakin, P. T., Priangan, N. R., & Blaang, A. A. D. (2025): *Implementasi algoritma AES-128 untuk pengamanan teks dalam sistem keamanan data*, Blantika: Multidisciplinary Journal, 3(6), 885–894.
- Hutasuhut, D. I. G., Aldizar, M. R., Nasution, I. F., & Nasution, M. F. (2023): *Perbandingan algoritma kriptografi simetris dan asimetris*, UNES Journal of Information System, 8(1), 42–47. <https://fe.ekasakti.org/index.php/UJIS>
- Novita, D., Zamzami, A., Anadi, Y. R., & Amini, S. (2025): *Mekanisme verifikasi data kependudukan oleh notaris terkait kebijakan hak akses penduduk*, Jurnal Dinamika Hukum, 31(2), 12211–12223.
- Nugraha, F., & Arifin, T. (2022): *Enkripsi dan dekripsi suara menggunakan metode AES 128b dengan secret key*, Sistemasi: Jurnal Sistem Informasi, 11(1), 97–107. <http://sistemasi.ftik.unisi.ac.id>
- Nuranisa, A., & Lukitasari, D. (2024): *Tindak pidana pencurian data dan privasi pengguna dalam transaksi e-commerce*, Amandemen: Jurnal Ilmu Pertahanan, Politik dan Hukum Indonesia, 1(2), 115–126. <https://doi.org/10.62383/amandemen.v1i2.145>
- Pradana, R. A. (2022): *Penerapan metode AES untuk keamanan repository dokumen digital di Pengadilan Militer III-13 Madiun*, Universitas Muhammadiyah Ponorogo.
- Prakoso, M. B., Wolor, C. W., & Marsofiyati. (2023): *Analisis pengelolaan dokumen di Departemen Pemasaran Kompas Gramedia*, Maeswara: Jurnal Riset Ilmu Manajemen dan Kewirausahaan, 1(6), 352–361. <https://doi.org/10.61132/maeswara.v1i6.420>
- Ramadhani, T. A., Cobantoro, A. F., & Sugianti. (2024): *Implementasi algoritma Advanced Encryption Standard 128 untuk pengamanan database sistem registrasi pasien*, JIP (Jurnal Informatika Polinema), 10(4), 521–526.
- Supriatiningsih. (2023): *Implementasi metode Waterfall pada aplikasi inventory barang berbasis web*, Informatics and Computer Engineering Journal, 3(2), 148–156. <https://jurnal.bsi.ac.id/index.php/ijec/>
- Susena, E., Puspitasari, N., & Faricha, N. (2023): *Sistem pengelolaan berkas berbasis website di Kantor Notaris-PPAT Wahid Nugroho Asri, S.H., M.Kn*, Jurnal INFORMA Politeknik Indonusa Surakarta, 9(1), 59–66.
- Tania, M., Alasi, T. S., & Yap, R. (2024): *Algoritma AES untuk keamanan data digital berbasis web di Kantor Desa Aman Damai*, Jurnal TIMES: Technology Informatics & Computer System, 13(2), 142–149. <http://ejournal.stmik-time.ac.id>
- Tarigan, Y. A. P., Aulia, R., & Elhanafi, A. M. (2024): *Algoritma AES 128 dalam mengenkripsikan berkas bansos Kecamatan Tigabinanga berbasis web*, Jurnal Unitek, 17(2), 193–204.
- Yuwana, M. A. S. A. S., Baihaqy, A. P. A., Fauzi, A. H. A., & Nur, M. A. (2025): *Analisa dampak kebocoran data Pusat Data Nasional (PDN) 2024 dalam perspektif HAM*, Jurnal Hukum dan HAM Wicarana, 4(1), 31–37.

Halaman ini sengaja dikosongkan