JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER

p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

Implementasi Keamanan Bukti Transaksi Medis Berbasis Web dengan Metode SHA-256 dan AES-128

Ahmad Yusril Ihsan Sya'bana[1*], Muhammad Afi Maulana[2], Muhlis Tahir[3]

[1,2 3]Fakultas Keguruan dan Ilmu Pendidikan, Pendidikan Informatika, Universitas Trunojoyo Madura, Bangkalan, Indonesia
Email: [1*]230631100008@student.trunojoyo.ac.id, [2*]230631100022@student.trunojoyo.ac.id, dan [3*]muhlis.tahir@trunojoyo.ac.id
(*Email Corresponding Author: 230631100008@student.trunojoyo.ac.id)
Received: June 9, 2026 | Revision: June 12, 2026 | Accepted: June 15, 2026

Abstrak

Penyimpanan dokumen digital berbasis web kini menjadi kebutuhan utama di berbagai institusi, namun risiko kebocoran data dan pencurian kredensial terus meningkat seiring maraknya serangan siber. Mengandalkan satu lapis pengamanan terbukti tidak lagi memadai karena jika satu titik pertahanan berhasil ditembus, seluruh data menjadi rentan. Penelitian ini membangun sistem manajemen dokumen berbasis web bernama SecureVault yang menerapkan arsitektur keamanan berlapis menggunakan kombinasi hashing SHA-256 dengan Salt untuk melindungi kata sandi pengguna, enkripsi AES-128 mode Cipher Block Chaining (CBC) untuk mengamankan dokumen yang diunggah, serta mitigasi kerentanan SQL Injection melalui Prepared Statements dan Cross-Site Scripting melalui sanitasi output htmlspecialchars(). Sistem dikembangkan menggunakan PHP dan MySQL dengan antarmuka berbasis Tailwind CSS. Pengujian dilakukan melalui empat skenario penetration testing dalam lingkungan server lokal. Hasil menunjukkan seluruh lapisan keamanan berfungsi dengan baik: mekanisme rate limiting berhasil memblokir akses setelah lima percobaan login gagal dengan waktu kunci 180 detik, tidak ada kata sandi yang tersimpan dalam bentuk plaintext di basis data, dokumen terenkripsi di server hanya menampilkan ciphertext yang tidak dapat dibaca, serta payload SQL Injection dan XSS berhasil dinetralisir sepenuhnya tanpa eksekusi. Integrasi antara kriptografi dan mitigasi kerentanan aplikasi menghasilkan sistem yang tangguh dan berlapis dalam mencegah kebocoran data.

Kata Kunci: SHA-256, AES-128, SQL Injection, Cross-Site Scripting, Keamanan Berlapis

Abstract

Web-based digital document storage has become a core need across institutions, yet risks of data breaches and credential theft continue to grow as cyber threats evolve. Relying on a single security layer is no longer sufficient since a breach at one point exposes the entire system. This study develops a web-based document management system called SecureVault, implementing a multi-layer security architecture that combines SHA-256 hashing with Salt to protect user passwords, AES-128 encryption in Cipher Block Chaining (CBC) mode to secure uploaded documents, SQL Injection mitigation through Prepared Statements, and Cross-Site Scripting prevention via htmlspecialchars() output sanitization. The system was built using PHP and MySQL with a Tailwind CSS interface. Testing was conducted through four penetration testing scenarios in a local server environment. Results show all security layers performed effectively: the rate limiting mechanism successfully blocked access after five consecutive failed login attempts with a 180-second lockout, no passwords were stored in plaintext within the database, encrypted files on the server displayed only unreadable ciphertext, and both SQL Injection and XSS payloads were fully neutralized without execution. The integration of cryptographic methods and application vulnerability mitigation produces a robust, layered system capable of significantly reducing the risk of data breaches in conventional web applications.

***Keywords**: SHA-256, AES-128, SQL Injection, Cross-Site Scripting, Multi-Layer Security*

1. PENDAHULUAN

Kemajuan teknologi saat ini mengharuskan banyak sektor, termasuk rumah sakit dan klinik, untuk menyimpan data penting secara *online* [12]. Meski sangat memudahkan, langkah ini mengundang bahaya besar dari serangan peretas (*hacker*). Salah satu data yang paling diincar dan harus dijaga kerahasiaannya adalah bukti transaksi medis [3]. Berbeda dengan nota belanja biasa, nota medis ini berisi informasi pribadi yang sangat sensitif, seperti riwayat penyakit, resep obat, jenis tindakan dokter, sampai data diri pasien. Menurut Undang-Undang Perlindungan Data Pribadi (UU PDP), data ini hukumnya wajib dirahasiakan. Jika sampai bocor, privasi pasien bisa terancam, identitas bisa disalahgunakan, dan pihak rumah sakit bisa terjerat kasus hukum.

Sayangnya, sistem keamanan web yang hanya mengandalkan satu lapis pertahanan sudah tidak lagi cukup [6]. Sering kali, pintu masuk para peretas justru berada pada halaman *login* karyawan. Menyimpan kata sandi (*password*) karyawan di dalam sistem penyimpanan (*database*) dalam bentuk teks biasa adalah kesalahan fatal [4].

Berbagai penelitian telah membuktikan bahwa ilmu persandian rahasia (kriptografi) adalah fondasi utama untuk melindungi data penting di dunia maya. Secara sederhana, kriptografi bekerja dengan cara menyamarkan data asli menjadi kode-kode acak yang tidak bermakna agar tidak bisa disadap oleh pihak yang tidak berwenang [15]. Untuk menyamarkan berkas dokumen fisik secara langsung saat disimpan (*data at rest*), penggunaan algoritma enkripsi simetris seperti

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

Advanced Encryption Standard (AES) varian 128-bit sangat direkomendasikan [14]. Metode ini ibarat brankas digital yang memotong data gambar ke dalam blok-blok berukuran 128-bit, lalu mengacaknya melalui proses matematika yang rumit. Agar hasil acakannya tidak membentuk pola berulang yang mudah ditebak oleh peretas, metode AES ini wajib dikombinasikan dengan sistem pengacakan berantai atau Mode Cipher Block Chaining (CBC) [10]. Mode CBC memastikan setiap potongan gambar diacak secara bergantung pada potongan sebelumnya, sehingga wujud aslinya benar-benar tersamarkan dan tidak dapat direkonstruksi tanpa kunci yang sah.

Sementara itu, untuk mengamankan identitas pengguna seperti kata sandi (*password*) di dalam pangkalan data (*database*), sistem menggunakan fungsi Secure Hash Algorithm (SHA-256). Metode ini bertugas layaknya mesin penghancur kertas yang mengubah kata sandi menjadi 256-bit potongan kode acak unik yang sifatnya permanen dan ireversibel (tidak bisa dikembalikan ke bentuk aslinya) [9]. Untuk mengantisipasi taktik peretas yang menebak sandi memakai daftar kamus otomatis (*dictionary attack*), kata sandi tersebut terlebih dahulu disisipi dengan Salt atau karakter acak tambahan sebelum diproses [1]. Dengan adanya sistem Salt ini, kata sandi yang sederhana sekalipun akan berubah menjadi kode yang sangat rumit dan mustahil untuk dibongkar [6].

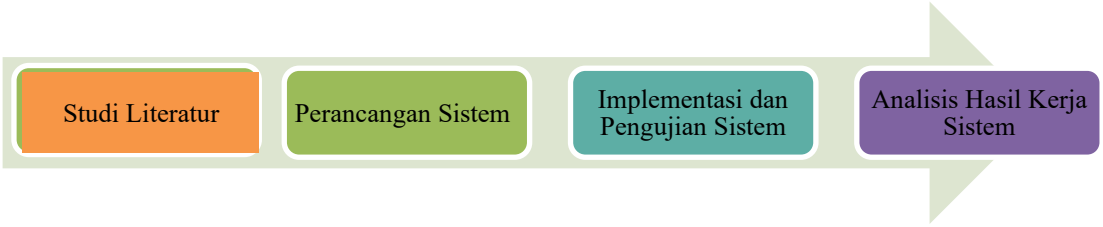
Namun, ilmu persandian kriptografi tidak akan bekerja maksimal jika aplikasi website-nya sendiri masih memiliki celah kerentanan. Sebagai langkah pencegahan agar peretas tidak bisa menyuntikkan perintah manipulasi ke dalam *database* (SQL Injection atau SQLi), sistem diwajibkan menggunakan teknik Prepared Statements. Selain itu, sistem juga perlu menetralkan karakter-karakter sensitif HTML menjadi entitas teks statis biasa untuk menangkal serangan skrip jahat (Cross-Site Scripting atau XSS). Langkah sterilisasi luaran ini memastikan peramban (*browser*) klien kebal dari jebakan program berbahaya saat mengakses website.

Berdasarkan *GAP Analysis* dari penelitian-penelitian sebelumnya terkait kerentanan sistem jika hanya menerapkan mekanisme perlindungan tunggal, maka penelitian ini bertujuan untuk membangun sistem keamanan web yang berlapis untuk melindungi bukti transaksi medis. Sistem ini menggabungkan pengunci dokumen (AES-128), pengacak kata sandi (SHA-256 dan Salt), serta penangkal celah website (anti SQLi dan XSS). Dengan konsep pertahanan berlapis (Defense in Depth) ini, diharapkan aset digital milik fasilitas kesehatan bisa benar-benar aman dari segala bentuk pencurian maupun manipulasi data [14].

2. METODOLOGI PENELITIAN

2.1 Metode Penelitian

Penelitian ini menggunakan pendekatan Research and Development (R&D) berbasis implementasi sistem, yang bertujuan untuk merancang, membangun, dan menguji arsitektur keamanan berlapis pada aplikasi web manajemen dokumen. Metodologi penelitian disusun secara sistematis ke dalam lima tahapan utama yang saling berkesinambungan, sebagaimana diilustrasikan pada Gambar 1.



Gambar 1. Contoh Alur tahapan penelitian

Tahap pertama dimulai dengan penelitian dilakukan dengan mengonfigurasi lingkungan server lokal menggunakan XAMPP yang mencakup Apache web server, PHP, dan MySQL., pada tahap selanjutnya dirancang arsitektur keamanan multi-layer yang mencakup empat lapisan perlindungan, kemudian sistem diimplementasikan menggunakan bahasa pemrograman PHP dengan framework CSS Tailwind untuk antarmuka pengguna dan MySQL sebagai sistem manajemen basis data. Implementasi mencakup empat komponen utama. Pengujian dilakukan menggunakan pendekatan penetration testing skala lokal terhadap empat skenario keamanan, Selanjutnya pada tahap analisis Hasil erja sistem diidentifikasi keterbatasan sistem dan rekomendasi pengembangan lebih lanjut, seperti penambahan Two-Factor Authentication (2FA). Data pengujian disiapkan dalam bentuk skenario penetration testing yang terdiri dari injeksi karakter bypass autentikasi, dokumen teks dan PDF, serta payload skrip JavaScript berbahaya. Prosedur pengujian dilakukan secara kotak hitam (black-box testing) dengan mensimulasikan serangan langsung pada antarmuka pengguna untuk mengamati respons sistem dan memvalidasi penanganan basis data terhadap input anomali.

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

2.2 Metode Implementasi dan Pengujian Sistem

Penelitian ini menerapkan dua metode kriptografi yang dikombinasikan untuk membentuk keamanan berlapis. Pemilihan metode didasarkan pada kajian komparatif oleh Dwiyansah et al. [8] yang membuktikan keunggulan AES-128 atas Blowfish dan varian AES lainnya dalam hal throughput pada sistem berbasis web.

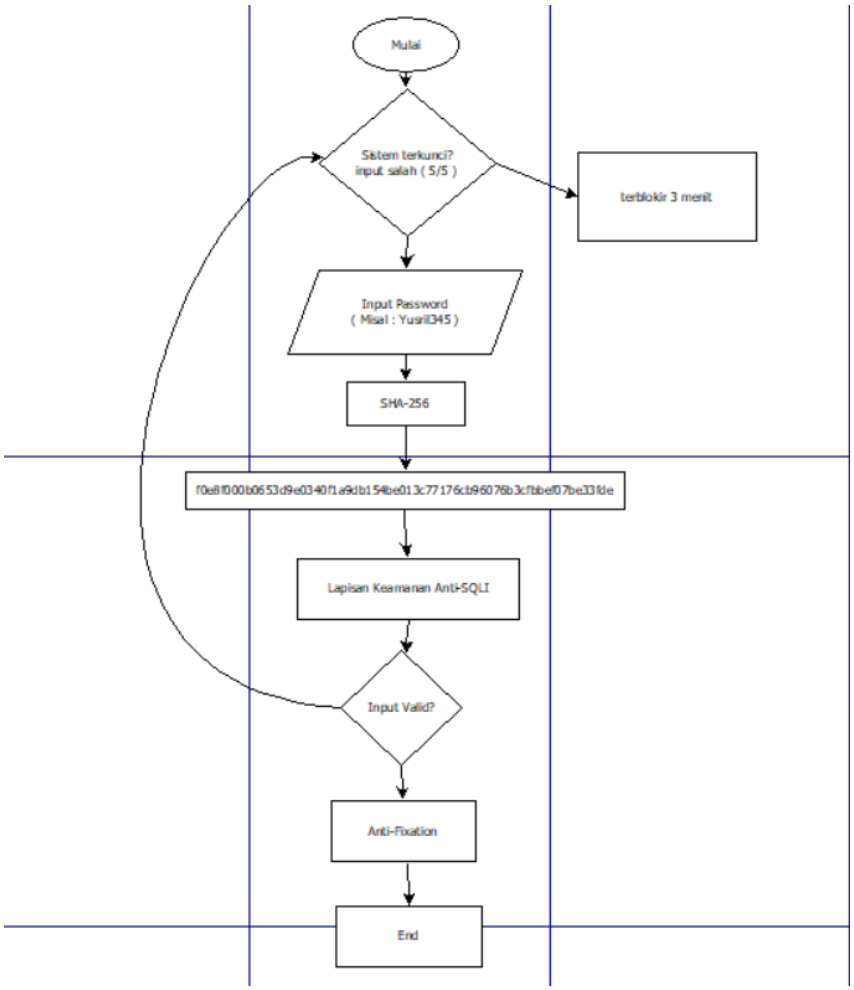
2.1.1 Metode Hashing SHA-256 dengan Salt pada Sistem Login

SHA-256 adalah fungsi hash satu arah yang menghasilkan digest sepanjang 256-bit. Algoritma ini bersifat deterministik dan collision-resistant, sehingga sangat ideal untuk proteksi kata sandi [9]. Untuk meningkatkan ketahanan terhadap *dictionary attack* dan rainbow table attack, nilai Salt acak 16-byte ditambahkan sebelum proses hashing dilakukan [2]. Rumus pembentukan hash kata sandi adalah sebagai berikut:

$$password\_hash = SHA\text{-}256(password\_asli \,||\, salt) \quad (1)$$

di mana || menyatakan operasi konkatenasi string, dan salt merupakan nilai acak yang unik bagi setiap pengguna. Nilai salt dan *password*_hash disimpan terpisah dalam basis data sehingga rekonstruksi kata sandi asli tidak dapat dilakukan meskipun basis data berhasil diakses oleh peretas [2].

Kemudian tahap-tahap implementasinya adalah sebagai berikut yang digambarkan dengan FlowChart :



Gambar 2. Proses Hashing pada Sistem Login

Pada proses ini menggambarkan alur sistem *login* dengan empat lapisan keamanan. Pertama, terdapat proteksi brute-force yang otomatis memblokir akses selama 3 menit jika pengguna salah menginput *password* sebanyak 5 kali. Kedua, *password* teks biasa yang dimasukkan langsung disamarkan menggunakan fungsi hash SHA-256 menjadi string acak 64 karakter demi menjaga kerahasiaan data. Ketiga, sebelum dilakukan validasi kecocokan data ke *database*, input dilewatkan melalui lapisan anti-SQLi untuk mencegah manipulasi query; jika data tidak valid, pengguna akan dikembalikan ke menu awal. Terakhir, jika *login* berhasil, sistem menerapkan fitur anti-fixation dengan memperbarui ID sesi (session regenerate) untuk mencegah pembajakan akun sebelum pengguna resmi masuk ke dalam sistem.

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

2.1.1 Metode Enkripsi AES-128 Mode CBC

AES-128 adalah algoritma kriptografi simetris standar internasional (NIST) yang memproses data dalam blok 128-bit menggunakan kunci 128-bit melalui 10 putaran komputasi [6]. Mode CBC dipilih karena kemampuannya menghasilkan Avalanche Effect yang tinggi: setiap blok plaintext di-XOR dengan blok ciphertext sebelumnya sebelum dienkripsi [10]. Mekanisme enkripsi CBC dirumuskan sebagai:

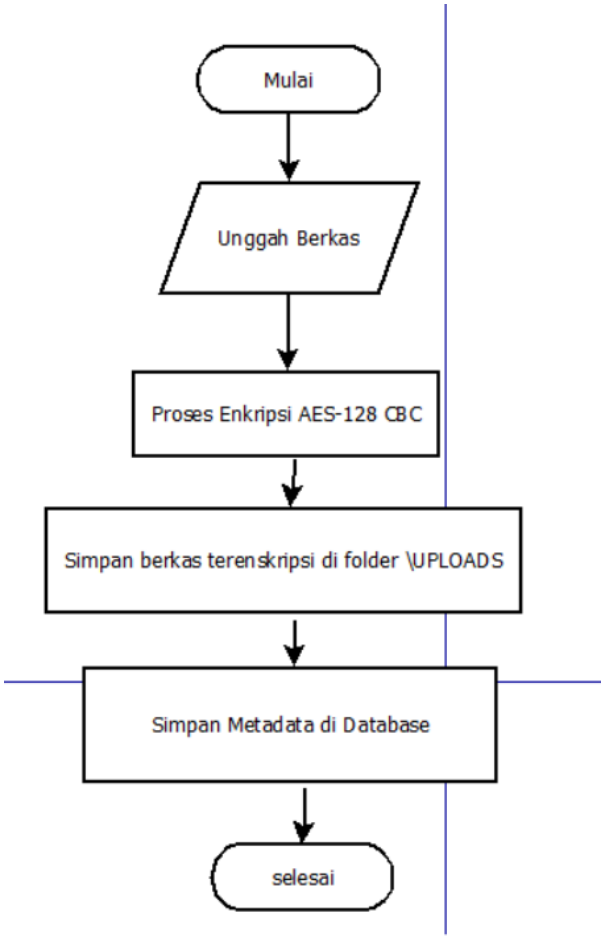
$$C\_i = AES\_Encrypt(P\_i \oplus C\_\{i\text{-}1\}) \quad (2)$$

$$C\_0 = IV \text{ (Initialization Vector acak 16-byte)} \quad (3)$$

di mana C_i adalah blok *ciphertext* ke-i, P_i adalah blok *plaintext* ke-i, dan $\oplus$ adalah operasi XOR. Berkat nilai IV yang dibangkitkan secara acak untuk setiap proses enkripsi, dua dokumen dengan isi identik akan menghasilkan *ciphertext* yang sepenuhnya berbeda [10]. Proses dekripsi CBC dirumuskan sebagai:

$$P\_i = AES\_Decrypt(C\_i) \oplus C\_\{i\text{-}1\} \quad (4)$$

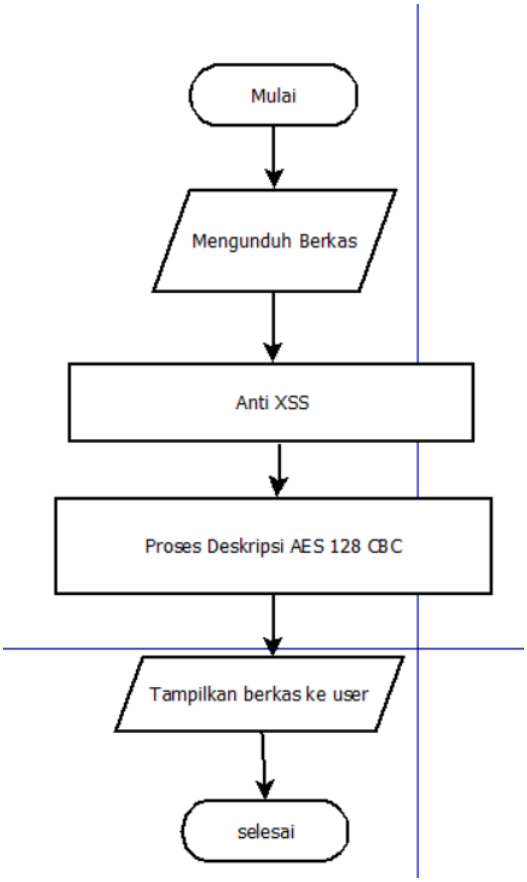
Kemudian tahap-tahap implementasinya adalah sebagai berikut yang digambarkan dengan FlowChart :



Gambar 3. Proses Enkripsi Berkas pada Dashboard

Proses Enkripsi dimulai saat pengguna mengunggah berkas ke dalam sistem, yang kemudian langsung diamankan menggunakan algoritma **AES-128 mode CBC** untuk mengubah isi berkas menjadi teks acak yang terenkripsi (*ciphertext*). Setelah itu, berkas yang telah terenkripsi disimpan secara fisik ke dalam direktori penyimpanan khusus bernama **\UPLOADS**. Terakhir, sistem mencatat informasi pendukung atau **metadata** berkas tersebut (seperti nama file, ukuran, atau parameter enkripsi) ke dalam *database* sebelum akhirnya seluruh rangkaian proses dinyatakan selesai.

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168



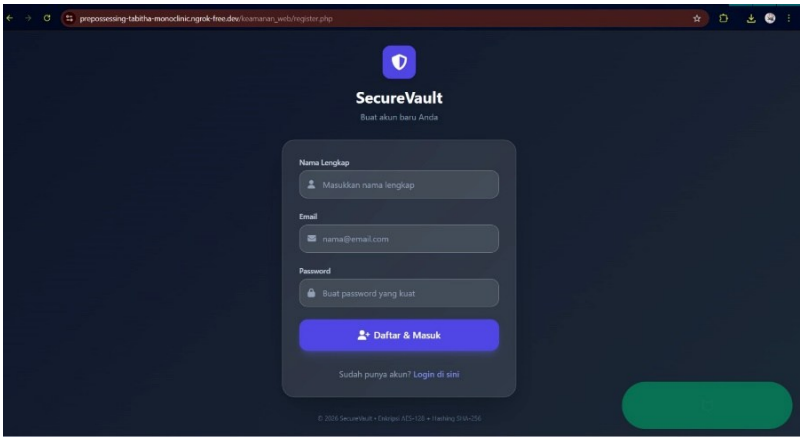
Gambar 4. Proses Deskripsi Berkas pada Dashboard

Pada Alur proses tersebut berrawal pengguna mengunduh berkas dari sistem. Sebelum berkas diproses lebih lanjut, sistem menerapkan filter Anti-XSS (Cross-Site Scripting) untuk memastikan parameter atau nama berkas tidak disisipi skrip berbahaya yang dapat menyerang peramban (*browser*) pengguna. Setelah divalidasi aman, berkas yang sebelumnya tersimpan dalam bentuk tersandi dikembalikan ke bentuk aslinya melalui Proses Dekripsi AES-128 CBC. Terakhir, berkas yang telah berhasil didekripsi ditampilkan atau diserahkan kepada pengguna dalam format asli sebelum akhirnya proses dinyatakan selesai.

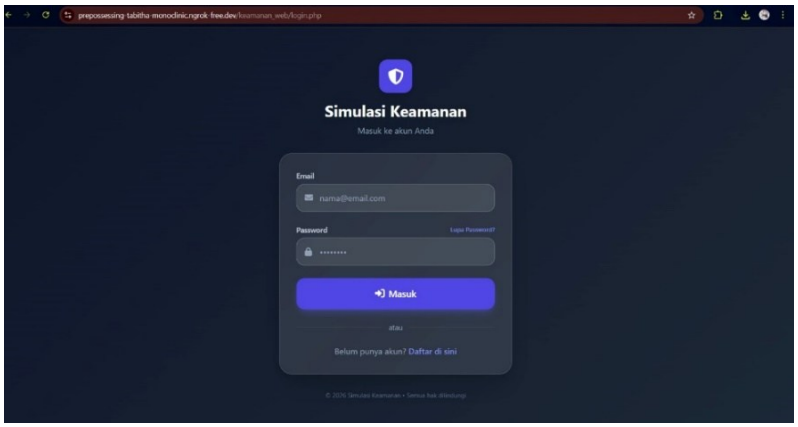
3. HASIL DAN PEMBAHASAN

Sistem web manajemen dokumen (SecureVault) telah berhasil diimplementasikan menggunakan bahasa pemrograman PHP dan sistem manajemen basis data MySQL. Antarmuka pengguna dibangun menggunakan framework Tailwind CSS sehingga menghasilkan tampilan yang responsif. Sistem ini memiliki dua antarmuka utama, yaitu halaman Autentikasi (Registrasi dan *Login*) serta halaman Dashboard Pengguna.

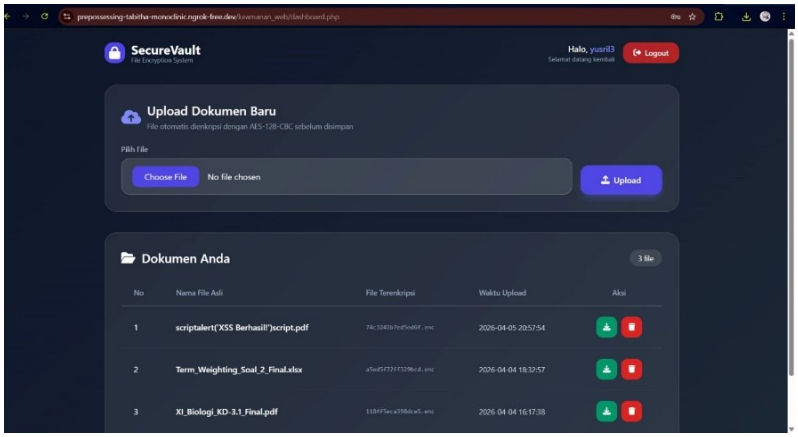
JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168



Gambar 5. Tampilan Halaman Daftar



Gambar 6. Tampilan Halaman Daftar



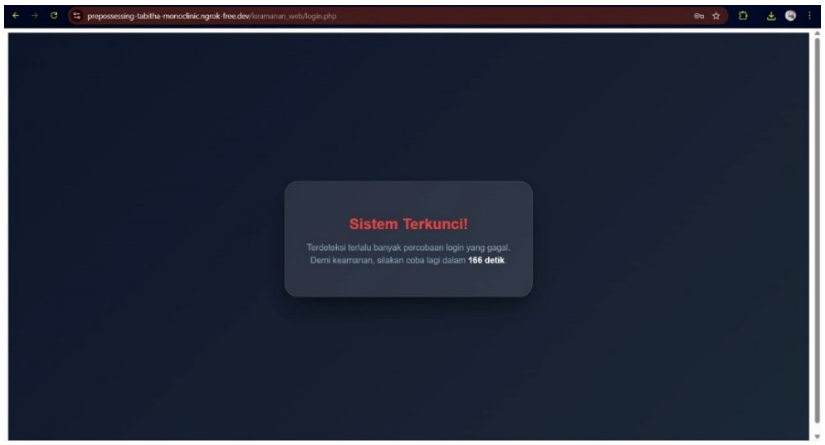
Gambar 7. Tampilan Halaman Dashboard

Berdasarkan ketiga gambar tersebut merupakan dari web yang kami buat, yang meliputi tampilan halaman daftar untuk mendaftarkan klien jika belum memiliki akun, halaman *login* untuk klien yang sudah memliki akun, halaman dashboard untuk mengunggah dan mengunduh berkas/

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

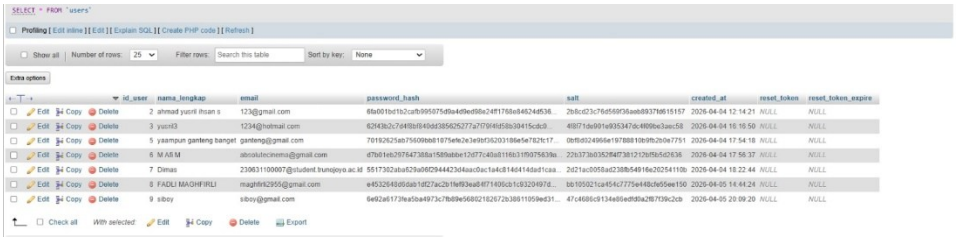
3.1 Pengujian Proteksi Brute-Force dan Kredensial

Pengujian pertama dilakukan pada halaman *Login* untuk menguji mekanisme pembatasan percobaan (rate limiting). Penguji dengan sengaja memasukkan kata sandi yang salah sebanyak 5 (lima) kali berturut-turut..



Gambar 8. Tampilan Halaman Dashboard yang terkunci

Berdasarkan gambar tersebut sistem secara otomatis memblokir akses pengguna sementara waktu dan menampilkan pesan "Sistem Terkunci!" dengan hitung mundur selama 180 detik (3 menit). Mekanisme ini terbukti efektif menggagalkan skenario serangan *brute-force* dan *dictionary attack.*

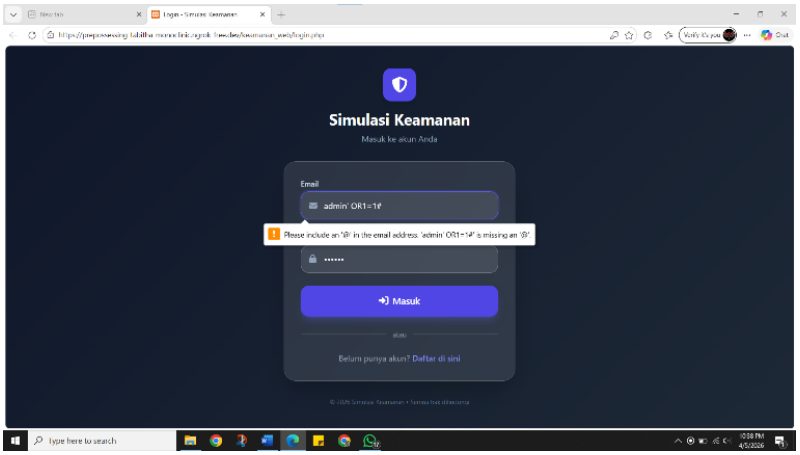


Gambar 9. Screenshot kolom *password*_hash dan kolom salt pada tabel users

Berdasarkan gambar tersebut pengecekan pada basis data menunjukkan bahwa kata sandi pengguna tidak disimpan dalam bentuk plaintext. Sistem murni menyimpan nilai Hash SHA-256 dan Salt acak sepanjang 16-byte, sehingga mencegah pencurian kredensial apabila basis data berhasil diretas.

3.2 Pengujian Mitigasi SQL Injection (SQLi)

Pengujian SQLi dilakukan pada form *Login* dengan memasukkan payload injeksi klasik seperti ' OR 1=1 -- pada kolom Email dan kata sandi acak.

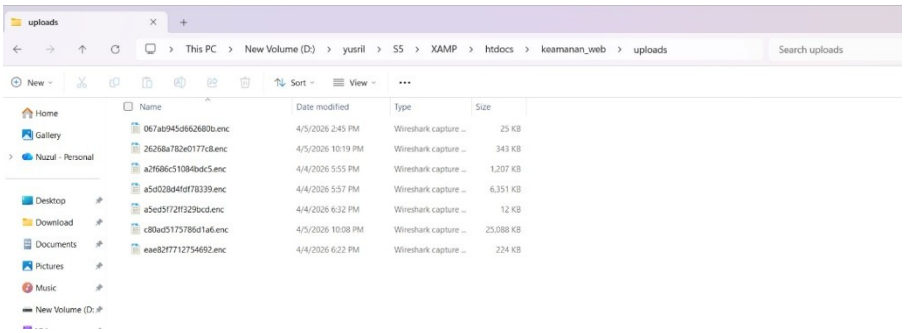


JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

Gambar 10. Pengujian SQL Injection

3.3 Pengujian Kriptografi AES-128 Mode CBC

Pengujian ini bertujuan untuk memastikan kerahasiaan dokumen yang diunggah ke server. Pengguna mengunggah sebuah dokumen teks asli (misalnya .pdf atau .txt).



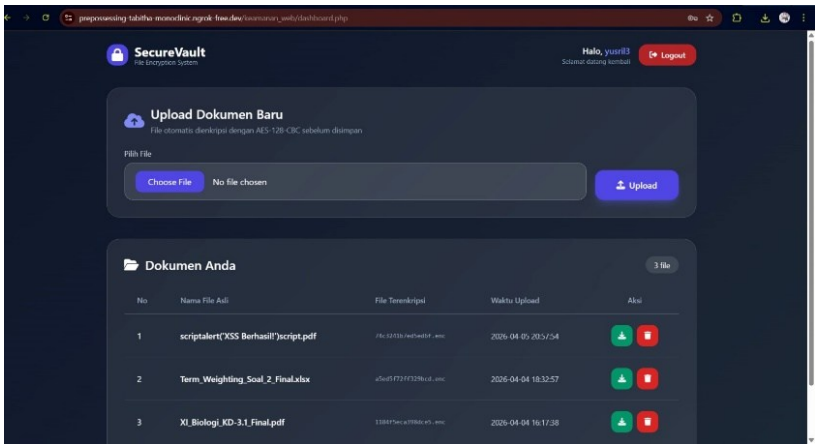
Gambar 11. Pengujian Kriptografi AES-128 Mode CBC

Pada Gambar tersebut Sistem berhasil mengeksekusi algoritma AES-128-CBC. Pada direktori fisik server (folder /uploads), file asli tidak ditemukan. Sistem menggantinya dengan file berekstensi .enc dengan nama acak. Saat file .enc tersebut dibuka secara paksa melalui text editor, isi file hanya menampilkan karakter ciphertext acak yang tidak dapat dibaca, membuktikan bahwa proses enkripsi berjalan sempurna. Selain itu, Sistem juga melakukan validasi kepemilikan dokumen saat proses dekripsi dan unduh (download). Jika file tersebut diunduh melalui akun yang sah, sistem berhasil mengembalikan file .enc menjadi dokumen aslinya secara utuh tanpa ada kerusakan data (corrupt).

3.1 Pengujian Mitigasi Cross-Site Scripting (XSS)

Pengujian XSS dilakukan dengan mengunggah file yang sengaja diberi nama menggunakan payload JavaScript, seperti <script>alert('XSS Berhasil!')</script>.pdf. Selain itu, mekanisme sanitasi menggunakan fungsi htmlspecialchars() diterapkan secara global pada seluruh output sistem, sehingga perlindungan terhadap serangan Cross-Site Scripting (XSS) juga berlaku pada seluruh form input lain seperti registrasi pengguna dan metadata dokumen.

Gambar 12. Notifikasi file berhasil di *upload*



Gambar 13. Pengujian Mitigasi Cross-Site Scripting (XSS)

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

Berdasarkan Gambar tersebut, saat sistem menampilkan daftar nama dokumen di halaman Dashboard, *browser* tidak mengeksekusi skrip tersebut (tidak muncul pop-up alert). Fungsi sanitasi htmlspecialchars() yang diterapkan pada kode program berhasil mengubah karakter HTML (tag) menjadi entitas biasa, sehingga skrip hanya ditampilkan sebagai teks literal dan terbukti aman dari kerentanan Stored XSS.

Tabel 1. Rekapitulasi Hasil Pengujian Keamanan Web

Skenario Pengujian	Modul Target	Payload / Input	Hasil Pengujian	Status
Brute-Force	Form Login	5x Salah Kata Sandi	Akun terkunci otomatis selama 180 detik	Berhasil
Keamanan Kredensial	Database	Pemeriksaan Tabel	Kata sandi tersimpan sebagai hash SHA-256 + Salt	Berhasil
SQL Injection	Form Login	OR 1=1 --	Payload ditolak dan tidak dieksekusi query	Berhasil
Enkripsi AES-128	Upload Berkas	Berkas PDF & Excel	Berkas menjadi file berekstensi .enc yang tidak terbaca	Berhasil
XSS	Dashboard	<script>alert('XSS')</script>	Skrip ditranslasikan menjadi teks statis (HTML Entities)	Berhasil

4. KESIMPULAN

Berdasarkan hasil Pengujian sistem manajemen dokumen SecureVault telah terbukti memenuhi standar keamanan arsitektur multi-layer yang kuat dan juga berhasil mengimplementasikan sistem manajemen dokumen berbasis web bernama SecureVault dengan arsitektur keamanan berlapis yang menggabungkan metode kriptografi dan mitigasi kerentanan aplikasi secara terintegrasi. Sistem dibangun menggunakan PHP dan MySQL, dengan antarmuka responsif berbasis Tailwind CSS, serta menerapkan empat pilar keamanan utama yang saling melengkapi. Penerapan algoritma hashing SHA-256 yang diperkuat dengan metode Salt terbukti efektif melindungi kredensial pengguna. Sistem tidak pernah menyimpan kata sandi dalam bentuk plaintext; setiap akun memiliki nilai Salt unik yang dibangkitkan secara kriptografis, sehingga serangan rainbow table dan *dictionary attack* tidak dapat menembus lapisan autentikasi. Mekanisme rate limiting yang memblokir akses selama 180 detik setelah lima percobaan *login* yang gagal turut memperkuat pertahanan terhadap serangan brute-force. Enkripsi dokumen menggunakan AES-128 mode Cipher Block Chaining (CBC) dengan Initialization Vector acak per sesi unggahan berhasil menjamin kerahasiaan fisik dokumen di server. File asli tidak tersimpan dalam folder uploads, melainkan digantikan oleh file terenkripsi berekstensi .enc yang hanya menampilkan ciphertext tidak bermakna jika dibuka secara paksa. Proses dekripsi berjalan sempurna tanpa korupsi data saat diakses oleh pengguna yang sah, sekaligus menolak akses dari akun yang tidak berhak. Mitigasi kerentanan tingkat aplikasi melalui Prepared Statements dan fungsi htmlspecialchars() berhasil menutup celah SQL Injection dan Cross-Site Scripting secara keseluruhan. Seluruh payload injeksi yang diujikan pada form *login* maupun nama dokumen diperlakukan sebagai teks biasa dan tidak dieksekusi oleh sistem maupun *browser*. Sinergi antara lapisan kriptografi dan lapisan mitigasi kerentanan aplikasi inilah yang menjadi keunggulan utama penelitian ini dibandingkan pendekatan keamanan tunggal, menghasilkan sistem yang tangguh, berlapis, dan siap dikembangkan lebih lanjut menuju lingkungan produksi skala penuh. Sebagai prospek pengembangan aplikasi lebih jauh, sistem manajemen dokumen ini dapat diintegrasikan dengan mekanisme autentikasi dua faktor (Two-Factor Authentication) untuk penguatan identitas berlapis, serta penambahan sistem deteksi intrusi untuk memantau lalu lintas aktivitas akses dokumen secara real-time

REFERENCES

[1] S. Mardhatillah, S. Winar, and D. Ammiady, "Optimasi Algoritma SHA-256 dan Metode Salt Untuk Pengamanan Akun Calon Santri Baru Pesantren Almuslim," Jurnal Ilmu Komputer Aceh, vol. 2, no. 3, 2025.

[2] P. D. Dwiyansah, F. Fathurrohman, N. Alfikry, and J. S. Asri, "A Comparative Analysis of the Advanced Encryption Standard (AES) 128-, 192-, and 256-Bit Algorithms in Digital Data Security," Jurnal Multidisiplin Sahombu, vol. 6, no. 02, pp. 176-182, 2026.

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 1 Juni 2026
Hal: 825-834
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i1.1168

[3] B. W. Widagdo, "Implementasi Aplikasi Enkripsi Data: Implementasi Aplikasi Enkripsi Data Menggunakan Algoritma AES 128 Untuk Meningkatkan Keamanan Informasi Berbasis Website di SMK Kusuma Bangsa," JUPIK: Jurnal Penelitian Ilmu Komputer, vol. 3, no. 1, pp. 16-26, 2025.

[4] F. Fadlullah, M. Tahir, B. P. Bintari, M. L. Dewi, M. F. Ilmy, and R. Ardiansyah, "Implementasi Algoritma AES pada Autentikasi Login Sistem Informasi," Jurnal Bintang Pendidikan Indonesia, vol. 1, no. 2, pp. 251-263, 2023.

[5] A. M. Ajif, F. Nuraeni, D. Kurniadi, and R. Elsen, "Implementasi Modul Tanda Tangan Digital dengan Superenkripsi RSA-ECDSA dan SHA-512 Pada Sistem Informasi Akademik Sekolah," Jurnal Algoritma, vol. 22, no. 2, pp. 933-944, 2025.

[6] F. M. Kaaffah, N. Nugroho, D. Nurnaningsih, and H. Harriansyah, "Sistem Enkripsi Dokumen Digital Melalui Kombinasi AES-128 dan Hashing SHA-256 Berbasis Salt," Jurnal Ilmiah Fifo, vol. 17, no. 1, pp. 78-90, 2025.

[7] A. Kautsar and M. Ikhsan, "Implementation of the Advanced Encryption Standard (AES) Algorithm and Bit Plane Complexity Segmentation (BPCS) Steganography Technique for Enhancing Text File Security," Sistemasi: Jurnal Sistem Informasi, vol. 14, no. 2, pp. 956-968, 2025.

[8] A. H. Wibowo and S. Sutardi, "Implementasi AES 256-bit untuk Enkripsi Dekripsi Teks dan Dokumen Word," Jurnal Informatika Ilmu Komputer dan Sistem Informasi, vol. 3, no. 3, 2025.

[9] Y. R. Hutasoit, V. Simangunsong, and D. Siallagan, "Analisis terhadap keamanan *password* menggunakan hash SHA-256," Jurnal Quancom: Quantum Computer Jurnal, vol. 3, no. 1, pp. 13-17, 2025.

[10] W. Prabowo and N. Anwar, "Pengujian Model Simulasi Efek Avalanche Kriptografi Simetris Algoritma AES 128-bit, Mode ECB dan CBC," IKRA-ITH Informatika: Jurnal Komputer dan Informatika, vol. 9, no. 1, pp. 178-186, 2025.

[11] F. D. Insani and M. D. Anggraeni, "Analisis Kinerja Algoritma Advanced Encryption Standard (AES) Encryption dan Algoritma Blowfish pada Proses Enkripsi dan Dekripsi," Jurnal TRANSFORMASI, vol. 21, no. 2, pp. 147-156, 2025.

[12] I. K. W. A. P. Ambara and B. G. K. Yudistira, "Analisis Komparatif Efektivitas Client-Side Encryption Cryptomator dan Relone Crypt pada Google Drive," Informatik: Jurnal Ilmu Komputer, vol. 21, no. 2, pp. 166-176, 2025.

[13] W. Rajiman and R. Indrayani, "Evaluasi Implementasi Kriptografi Metode AES-256 Bit dan Fungsi Hash 256 Bit Sebagai Upaya Pengamanan Dokumen," Explore, vol. 15, no. 2, pp. 121-129, 2025.

[14] M. B. Aryanto, M. Tahir, S. I. Devita, Z. N. Mustofa, Q. Ainiyah, dan S. Sundoro, "Implementasi enkrip dan dekrip file menggunakan metode advance encryption standard (AES-128)," J. Ilm. Sist. Inf. dan Ilmu Komput, vol. 3, no. 1, pp. 89-104, 2023.

[15] B. O. Irawan, M. Tahir, N. A. Windrastuti, D. Y. Cholili, D. Mulaikah, and A. B. M. S. Wachid, "Implementasi Kriptografi Pada Keamanan Data Menggunakan Algoritma Advance Encryption Standard (Aes)," Jurnal Simantec, vol. 11, no. 2, pp. 167-174, 2023.