

APLIKASI WEB KEAMANAN DOKUMEN MENGGUNAKAN RIVEST- SHAMIR-ADLEMAN DAN ADVANCED ENCRYPTION STANDARD

Muhamad Ridwan¹, Tati Ernawati²

Politeknik TEDC Bandung¹²

Jl. Pesantren, Cibabat, Kec. Cimahi Utara, Kota Cimahi, Jawa Barat 40513

E-mail: mrdwn1812@gmail.com¹, tatiernawati@poltektedc.ac.id²

ABSTRAK

Pengelolaan dokumen yang efektif di lingkungan sekolah menjadi hal penting untuk menunjang kelancaran operasional dan administrasi. Metode penyimpanan konvensional berbasis kertas masih umum digunakan, namun memiliki berbagai keterbatasan, terutama dalam hal kebutuhan ruang dan risiko kerusakan fisik. Seiring kemajuan teknologi informasi, pendekatan *paperless* menjadi alternatif yang lebih efisien. Meski demikian, penggunaan dokumen digital memunculkan tantangan baru terkait keamanan data. Penelitian ini merancang dan membangun aplikasi berbasis web menggunakan framework Laravel dan database MariaDB, dengan mengimplementasikan algoritma enkripsi RSA dan AES guna meningkatkan keamanan dokumen digital, khususnya soal-soal ujian sekolah. Metode pengembangan sistem yang digunakan adalah model waterfall, dengan tahapan yang sistematis dari analisis hingga pemeliharaan. Hasil implementasi menunjukkan bahwa sistem mampu menyimpan dan mengelola dokumen secara efisien, mendukung akses multiuser, dan menjaga kerahasiaan data melalui fitur enkripsi. Dengan adanya aplikasi ini, sekolah dapat bertransisi ke sistem penyimpanan digital secara aman dan terstruktur.

Kata kunci: Pengelolaan dokumen, Keamanan data, *Paperless*, RSA, AES, Laravel.

ABSTRACTS

Effective document management in a school environment is essential to support smooth operations and administration. Conventional paper-based storage methods are still commonly used but come with various limitations, particularly in terms of space requirements and the risk of physical damage. With the advancement of information technology, a paperless approach has become a more efficient alternative. However, the use of digital documents introduces new challenges related to data security. This study designs and develops a web-based application using the Laravel framework and MariaDB database, implementing RSA and AES encryption algorithms to enhance the security of digital documents, specifically school exam papers. The system development method used is the waterfall model, with systematic stages from analysis to maintenance. The implementation results show that the system can store and manage documents efficiently, support multi-user access, and maintain data confidentiality through encryption features. With this application, schools can transition to a secure and well-structured digital storage system.

Keywords: Document management, Data security, Paperless, RSA, AES, Laravel.

1. PENDAHULUAN

Pengelolaan dokumen di lingkungan sekolah memiliki peran penting dalam menunjang kelancaran operasional dan administrasi. Penyimpanan dokumen dalam bentuk fisik (*paper based*) memang masih banyak digunakan, namun metode ini memiliki keterbatasan, salah satunya adalah kebutuhan ruang penyimpanan yang besar. Seiring perkembangan teknologi informasi, penyimpanan dokumen dalam bentuk *softcopy* (*paperless*) menjadi alternatif yang lebih efisien dan praktis.

Dokumentasi *paperless* memberikan banyak keuntungan seperti efisiensi ruang dan kemudahan akses, di sisi lain muncul permasalahan baru terkait keamanan data. Dokumen digital yang tidak

dilindungi dengan baik berpotensi diakses oleh pihak yang tidak berwenang dan disalahgunakan untuk tujuan yang tidak sah [1]. Berdasarkan laporan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) tahun 2024, kasus pencurian data pribadi di Indonesia mengalami peningkatan signifikan sebesar 20,97% dibandingkan tahun sebelumnya yang hanya 7,96% [2].

Permasalahan keamanan data ini dapat membawa dampak besar, seperti kebocoran informasi sensitif, kerugian finansial, dan hilangnya kepercayaan terhadap lembaga terkait. Selain itu, data yang tidak terlindungi juga rentan terhadap serangan *malware* dan peretasan [3]. Menurut [1], pengamanan dokumen sangat penting untuk

melindungi data sensitif dan rahasia, sedangkan [4] menegaskan bahwa kehilangan data dapat menyebabkan kerugian besar bagi organisasi.

Mengingat pentingnya pengamanan dokumen, diperlukan sebuah solusi yang tidak hanya mendukung transisi dari *paper based* ke *paperless*, tetapi juga menjamin keamanan dokumen yang tersimpan secara digital. Salah satu metode efektif dalam menjaga keamanan dokumen adalah dengan menerapkan algoritma enkripsi, seperti Rivest-Shamir-Adleman (RSA) dan *Advanced Encryption Standard* (AES). RSA merupakan algoritma kriptografi yang menggunakan metode kunci publik dan kunci privat untuk enkripsi dan dekripsi data, menawarkan tingkat keamanan yang tinggi. Sedangkan AES digunakan untuk mengenkripsi teks maupun file dalam ukuran besar, menggunakan enkripsi simetris berbasis *ciphertext* [5]. Dengan mengkombinasikan kedua algoritma ini, diperoleh keamanan data yang lebih optimal dan fleksibel [6].

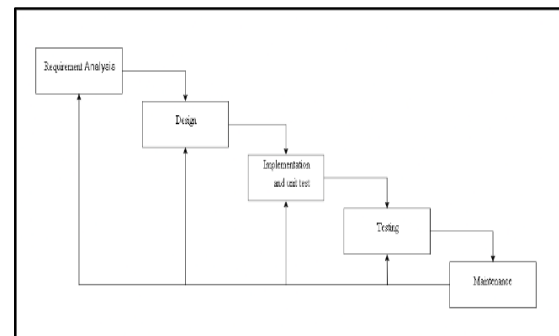
Penulis melakukan observasi pada tanggal 11 Januari 2025 di Madrasah Ibtidaiyah (MI) Nurul Falah Cibuluh, Kabupaten Subang. Berdasarkan hasil wawancara dengan pihak sekolah, diketahui bahwa manajemen dokumen masih dilakukan secara manual (*paper based*), yang mengakibatkan keterbatasan ruang dokumentasi. Pihak sekolah juga menyatakan keinginan untuk beralih ke sistem *paperless*, namun masih memiliki kekhawatiran terkait keamanan dokumen digital.

Berdasarkan permasalahan tersebut, penulis mengusulkan pembuatan aplikasi berbasis web menggunakan Framework Laravel versi 9.19 dan database MariaDB versi 11.6, yang mengimplementasikan kombinasi algoritma RSA dan AES untuk pengamanan dokumen, khususnya dokumen soal-soal ujian sekolah. Dengan dikembangkannya aplikasi ini, diharapkan pihak sekolah dapat mengelola dokumen secara digital dengan aman, mengurangi kebutuhan ruang fisik untuk penyimpanan dokumen, meningkatkan perlindungan terhadap data penting, serta memudahkan pengelolaan dan akses dokumen dengan kontrol keamanan yang lebih baik. Penelitian ini dituangkan dalam laporan Tugas Akhir berjudul “Aplikasi Pengamanan Dokumen dengan Algoritma Rivest Shamir Adleman dan *Advanced Encryption Standard* (Studi Kasus Madrasah Ibtidaiyah Nurul Falah Cibuluh Subang)”.

2. METODE PENELITIAN

System Development Life Cycle atau yang lebih dikenal dengan istilah SDLC adalah metodologi umum yang digunakan untuk mengembangkan sistem informasi. Terdapat beberapa model pendekatan ini, model yang paling sering digunakan adalah model *waterfall* dan model

prototype. Penulis menggunakan model *waterfall* dalam kajian penelitian ini. Model *waterfall* merupakan salah satu model yang sering digunakan atau sering disebut dengan model regular atau siklus hidup klasik pada model SDLC. Model ini menggunakan pendekatan sistematis dan berurutan, mulai dari tingkat kebutuhan sistem dan berlanjut ke analisis, desain, pengkodean, pengujian/validasi, dan pemeliharaan [7]. *Waterfall* memuat beberapa tahapan yang sistematis dan berurut dimulai dari requirements analysis, design, implementation, testing, dan maintenance [8]. Ilustrasi dari SDLC *Waterfall* dapat dilihat pada Gambar 1.1.



Gambar 1.1 Metode Pengembangan *Waterfall*

1. Requirement Analysis

Tahap awal dalam proses perancangan aplikasi adalah melakukan analisis kebutuhan, yang bertujuan untuk mengidentifikasi secara jelas kebutuhan pengguna serta batasan sistem yang akan dikembangkan. Pada tahap ini, penulis melakukan pengumpulan data melalui metode observasi dan wawancara langsung kepada beberapa pihak di lokasi studi kasus, yaitu MI Nurul Falah Cibuluh. Proses ini penting untuk memahami permasalahan yang dihadapi oleh pihak sekolah dan untuk merumuskan kebutuhan sistem secara tepat, seperti siapa pengguna aplikasinya serta fitur-fitur apa saja yang dibutuhkan.

Analisis kebutuhan menjadi dasar dalam membangun aplikasi yang sesuai dengan harapan pengguna dan menjadi jembatan antara pengguna dan tim pengembang. Menurut [9], “*Requirement analysis focuses on identifying what the system should do, the services it provides, and the constraints under which it operates*” Selain itu, menurut [10], “*Requirements determination is the most critical step in the systems development process, as errors made in this phase can lead to costly fixes later*”.

2. Design

Tahap desain merupakan fase penting dalam pengembangan perangkat lunak yang bertujuan menerjemahkan kebutuhan pengguna yang telah dianalisis ke dalam bentuk rancangan sistem yang

terstruktur. Pada tahap ini, dibuat blueprint atau cetak biru sistem yang berfungsi sebagai panduan teknis pembangunan aplikasi. Desain ini mencakup struktur data, arsitektur sistem, antarmuka pengguna, serta alur proses yang sesuai dengan kebutuhan fungsional maupun non-fungsional.

Penulis merancang sistem menggunakan pendekatan berbasis objek dengan *Unified Modeling Language* (UML) yang meliputi *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*. UML dipilih karena kemampuannya untuk memvisualisasikan sistem secara menyeluruh dan mendukung dokumentasi yang jelas serta mudah dipahami oleh seluruh pemangku kepentingan. Selain itu, menurut *Object Management Group* (OMG, 2023), "*UML is a standardized modeling language that enables developers to specify, visualize, and document models of software systems, including their structure and behavior*".

3. Implementation and unit test

Tahap implementasi merupakan proses penerjemahan rancangan sistem ke dalam kode program nyata menggunakan bahasa pemrograman dan alat bantu pengembangan perangkat lunak. Pada tahap ini, pengembang mulai menulis kode berdasarkan desain yang telah disusun sebelumnya, membangun modul-modul kecil secara terpisah yang nantinya akan diintegrasikan dalam tahap pengujian dan integrasi.

Perangkat lunak yang digunakan dalam proses implementasi ini antara lain *Visual Studio Code* sebagai code editor, *MariaDB* sebagai *database management system*, dan web browser untuk menampilkan serta menguji antarmuka aplikasi. Bahasa pemrograman yang digunakan adalah PHP, dengan menggunakan *Laravel*, salah satu *framework* modern yang mendukung pengembangan aplikasi web secara cepat dan terstruktur.

Menurut [11], "*PHP combined with frameworks like Laravel allows for rapid application development with built-in features for routing, templating, and database management*." Sementara itu, *Laravel* sendiri terus berkembang dan dalam dokumentasi resminya disebutkan sebagai "*a web application framework with expressive, elegant syntax designed to ease common tasks used in most web projects*." [12].

4. Testing

Tahap pengujian merupakan langkah penting untuk memastikan bahwa aplikasi yang dikembangkan berjalan sesuai dengan desain, memenuhi

kebutuhan fungsional, serta bebas dari kesalahan logika atau bug. Pengujian ini dilakukan untuk mengevaluasi apakah sistem telah berfungsi sebagaimana mestinya dan siap untuk digunakan oleh pengguna akhir.

Metode yang digunakan dalam pengujian ini adalah *Black Box Testing*, yaitu metode pengujian fungsional yang berfokus pada input dan output tanpa mengetahui struktur internal kode program. Pengujian ini berguna untuk memastikan bahwa setiap fitur dalam aplikasi bekerja sesuai harapan pengguna.

Selain itu, dilakukan pula *Acceptance Testing* dengan pendekatan *User Acceptance Test* (UAT) atau sering disebut *Beta Testing*, yaitu pengujian langsung oleh pengguna akhir untuk menilai apakah aplikasi sudah memenuhi kebutuhan bisnis dan dapat digunakan dalam lingkungan nyata.

Menurut [13], "*Black-box testing is essential for validating software functions without bias from internal implementations*".

5. Maintenance

Tahap *maintenance* merupakan bagian penting dari siklus hidup perangkat lunak, yang dilakukan setelah sistem di-deploy ke lingkungan produksi. Tahapan ini mencakup aktivitas pemeliharaan seperti perbaikan kesalahan (*bug fixing*), peningkatan kinerja, pembaruan fitur, serta penyesuaian sistem terhadap kebutuhan pengguna atau perubahan lingkungan teknologi.

Dalam pengembangan sistem ini, penulis belum melaksanakan tahap *maintenance* karena sistem belum di-deploy ke server produksi dan masih dalam tahap pengujian internal.

Menurut [14], "*Software maintenance refers to the process of modifying a software product after it has been delivered to correct faults, improve performance, or adapt it to a changed environment*." Selain itu, menurut [15], "*Software maintenance is the modification of a software product after delivery to correct defects, to improve performance or other attributes, or to adapt the product to a modified environment*".

3. HASIL DAN PEMBAHASAN

Pada tahap ini, dipaparkan hasil dari proses analisis dan perancangan sistem keamanan dokumen yang telah dibuat. Proses diawali dengan pengumpulan kebutuhan fungsional pengguna melalui metode wawancara langsung, guna mendapatkan gambaran yang jelas tentang fitur dan fungsi yang diharapkan dalam sistem. Setelah kebutuhan tersebut

dikumpulkan, langkah berikutnya adalah memvisualisasikannya dalam bentuk *Use Case Diagram*. Diagram ini berfungsi untuk memperjelas interaksi antara aktor pengguna dengan berbagai fungsi yang tersedia dalam sistem, sehingga memudahkan pemahaman dan perancangan lebih lanjut.

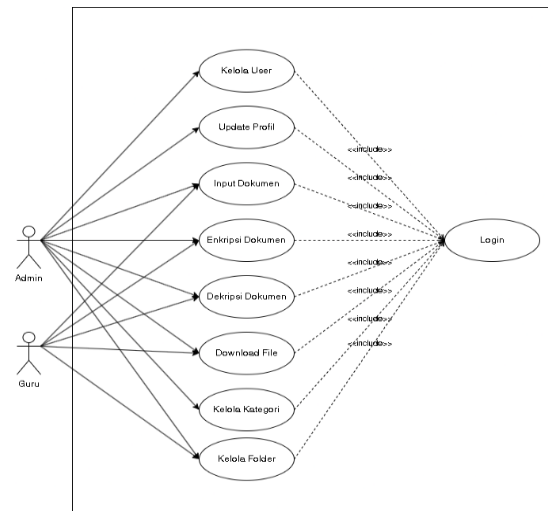
Implementasi sistem dilakukan menggunakan framework Laravel, yang dipilih karena kemampuannya dalam membangun aplikasi web yang aman dan efisien. Setelah tahap pengembangan selesai, dilakukan serangkaian pengujian untuk mengevaluasi kinerja dan memastikan bahwa semua fitur berjalan sesuai dengan spesifikasi yang telah ditetapkan sebelumnya. Pengujian ini bertujuan untuk mendeteksi dan memperbaiki kemungkinan bug atau kesalahan, sehingga hasil akhir sistem dapat beroperasi dengan optimal dan memenuhi kebutuhan pengguna secara menyeluruh.

3.1. Kebutuhan Pengguna

Setelah dilakukannya wawancara yang dilakukan oleh penulis dengan pihak MI Nurul Falah Cibuluh pada tanggal 20 desember 2024 sampai dengan 21 Januari 2025. Penulis memperoleh beberapa *user requirements* untuk keberlangsungan perancangan aplikasi pengamanan dokumen diantaranya adalah sebagai berikut:

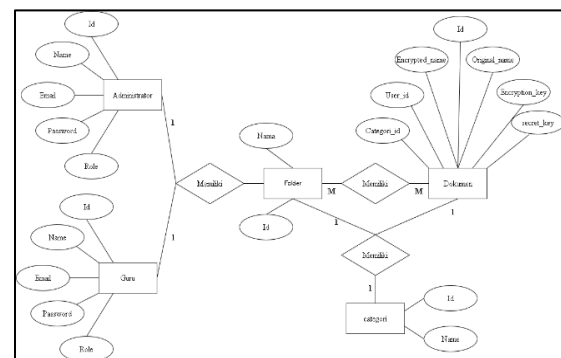
1. Sistem ini dirancang untuk mempermudah kegiatan penyimpanan dan keamanan dokumen yang ada serta proses pencarian dokumen di MI Nurul Falah.
 2. Sistem ini dapat digunakan *user* diantaranya adalah guru dan staff tata usaha sekolah.
 3. Sistem ini memiliki fitur manajemen *user* yang dapat diakses oleh administrator dan kepala sekolah. Admin dapat melakukan Operasi *Create, Read, Update, Delete* (CRUD).
 4. Sistem ini memiliki fitur manajemen dokumen. *User* dapat melakukan Operasi *Create, Read, Update, Delete* (CRUD).
- Sistem ini memiliki fitur laporan yang mendaftarkan semua dokumen.

3.2. Use Case Diagram



Gambar 3.1 Use Case Diagram Sistem

3.3. Entity Relational Diagram

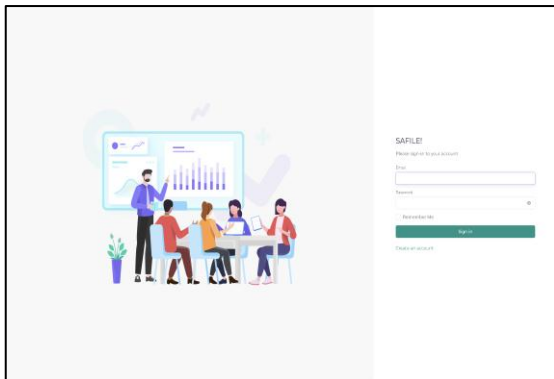


Gambar 3.2 Entity Relational Diagram

Gambar 3.2 menggambarkan bahwa sistem pengamanan dokumen memiliki 4 entitas diantaranya adalah:

- a. Administrator dengan atribut *id, name, email, password*, dan *role*.
- b. Guru dengan atribut *id, name, email, password*, dan *role*.
- c. Dokumen dengan atribut *id, user_id, category_id, encrypted name, original name, encrypted key* dan *secret key*.
- d. Status dengan atribut *id* dan *name*.
- e. Category dengan atribut *id*, dan *name*.
- f. Folder dengan atribut *id* dan *name*.

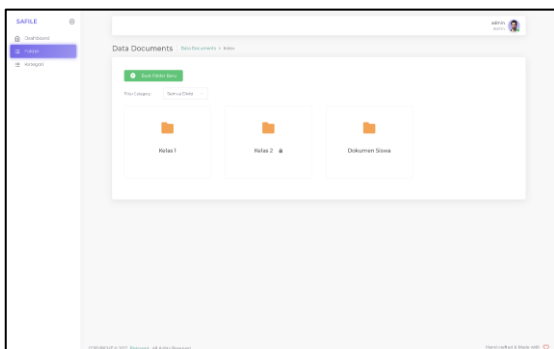
3.4. Halaman Login



Gambar 3.2 Halaman Login

Gambar 3.2 menampilkan halaman login yang merupakan pintu masuk utama ke dalam sistem. Pada tampilan ini, terdapat nama sistem yang ditampilkan secara jelas sebagai identitas aplikasi, disertai dengan dua field input, yaitu *Email* dan *Password*. Pengguna diwajibkan untuk mengisi kedua field tersebut menggunakan data akun yang telah terdaftar sebelumnya di dalam sistem. Proses ini bertujuan untuk memastikan bahwa hanya pengguna yang memiliki kredensial yang valid yang dapat mengakses fitur dan data yang tersedia, sehingga keamanan dan privasi informasi dalam sistem tetap terjaga dengan baik.

3.5. Halaman Folder

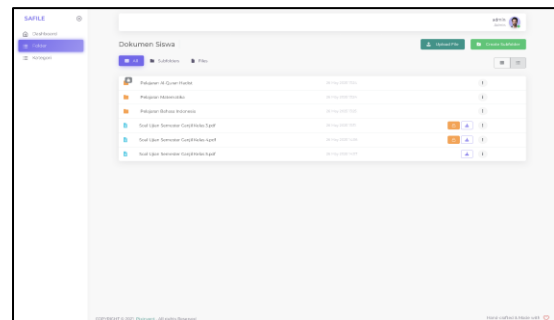


Gambar 3.3 Halaman Folder

Gambar 3.3 memperlihatkan antarmuka menu folder yang menyajikan daftar folder yang telah disimpan dalam sistem. Menu ini dirancang untuk memudahkan pengguna dalam mengelola struktur penyimpanan, sehingga setiap folder yang telah dibuat atau diunggah oleh pengguna dapat ditampilkan secara terorganisir. Tampilan ini juga berfungsi sebagai pusat navigasi, memungkinkan pengguna untuk membuka, memeriksa isi, atau melakukan tindakan lanjutan seperti mengedit dan

menghapus folder yang tersedia. Dengan adanya menu folder ini, sistem menjadi lebih *user-friendly* dan efisien dalam membantu pengguna mengakses data yang tersimpan.

3.6. Halaman Detail Folder



Gambar 3.4 Halaman Detail Folder

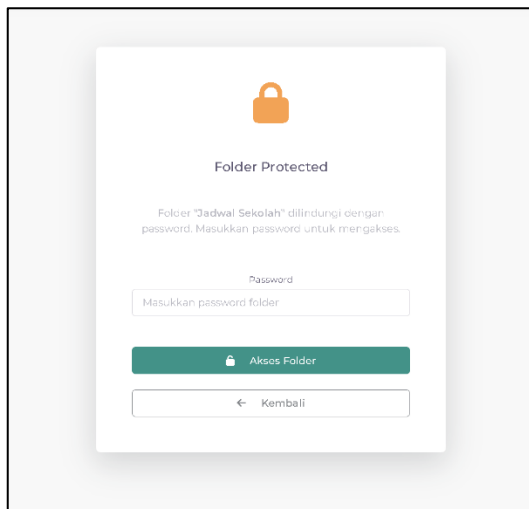
Gambar 3.4 menunjukkan tampilan antarmuka detail dari sebuah folder, di mana pengguna dapat melihat daftar folder di dalamnya beserta semua file yang telah tersimpan dalam sistem. Tampilan ini memberikan gambaran yang lebih rinci mengenai isi dari folder yang dipilih, sehingga pengguna dapat dengan mudah menavigasi struktur data yang bersifat hierarkis. Setiap subfolder dan file ditampilkan secara terorganisir untuk memudahkan proses pengelolaan, pencarian, serta tindakan lainnya seperti membuka, mengunduh, mengedit, atau menghapus. Fitur ini sangat membantu dalam mendukung efisiensi kerja, terutama saat pengguna perlu mengakses atau memanipulasi berbagai berkas dan folder secara cepat dan terstruktur.

3.7. Halaman Tambah Folder

Gambar 3.5 Halaman Tambah Folder

Gambar 3.5 memperlihatkan tampilan antarmuka yang digunakan untuk proses pembuatan folder baru dalam sistem. Pada tampilan ini, pengguna disediakan beberapa field input seperti nama folder untuk menentukan identitas folder, kategori folder yang berfungsi untuk mengelompokkan folder sesuai jenis atau fungsinya, serta *password* yang bersifat opsional dan dapat digunakan untuk memberikan perlindungan tambahan terhadap folder tersebut. Setelah seluruh data diisi dengan benar, pengguna dapat menekan tombol *Create Folder* yang berfungsi untuk menyimpan informasi folder ke dalam sistem. Fitur ini dirancang untuk memberikan fleksibilitas dan keamanan dalam pengelolaan data, sekaligus menjaga keteraturan struktur folder yang ada di dalam sistem.

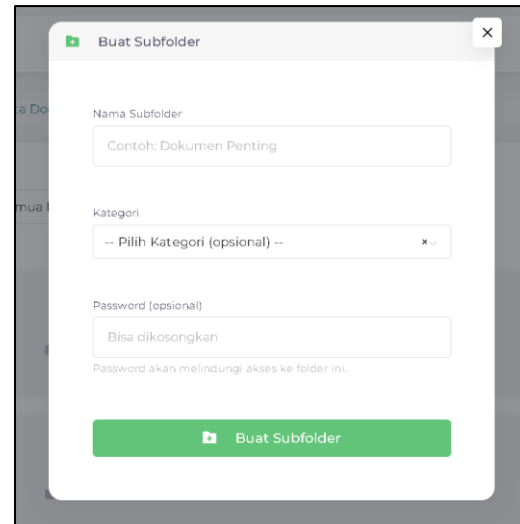
3.8. Halaman Buka Password Folder



Gambar 3.6 Halaman Membuka *Password Folder*

Gambar 3.6 memperlihatkan tampilan antarmuka yang digunakan untuk mengakses folder atau subfolder yang dilindungi oleh sandi. Dalam tampilan ini, pengguna diminta untuk memasukkan password pada field yang telah disediakan. Setelah sandi dimasukkan dengan benar, tombol *Buka Folder* dapat ditekan untuk membuka dan menampilkan isi dari folder tersebut. Mekanisme ini dirancang untuk menjaga keamanan data yang tersimpan, sehingga hanya pengguna yang memiliki akses resmi yang dapat melihat atau mengelola konten di dalamnya. Fitur ini memberikan lapisan perlindungan tambahan terhadap informasi penting yang terdapat dalam sistem.

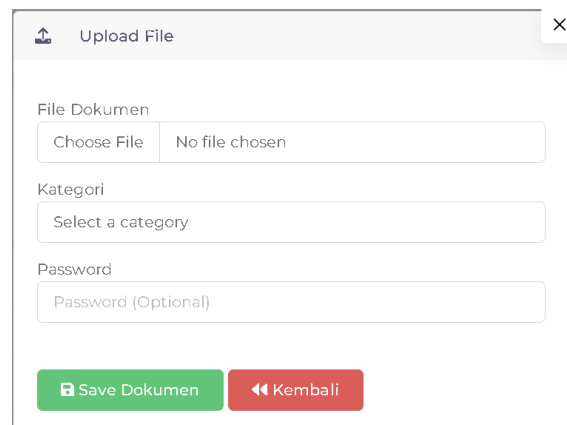
3.9. Halaman Upload Dokumen



Gambar 3.7 Halaman untuk Membuat Subfolder

Gambar 3.7 memperlihatkan antarmuka yang digunakan untuk membuat subfolder di dalam sistem. Pada tampilan ini, tersedia beberapa isian penting, yaitu nama folder untuk menentukan identitas subfolder yang akan dibuat, serta *password* opsional yang dapat digunakan untuk memberikan perlindungan tambahan terhadap akses folder tersebut. Setelah seluruh informasi diisi, pengguna dapat menekan tombol *Create Folder* guna menyimpan data subfolder ke dalam sistem. Fitur ini dirancang untuk mendukung pengelompokan data secara bertingkat, sehingga struktur penyimpanan menjadi lebih rapi, aman, dan mudah dikelola.

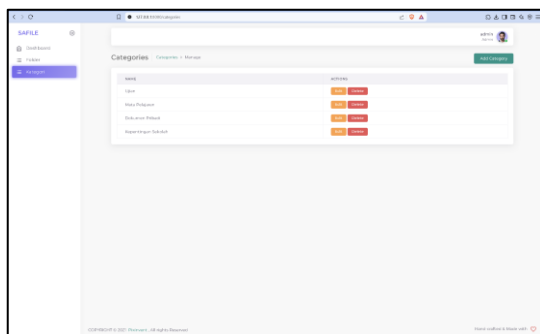
3.10. Halaman Upload File



Gambar 3.8 Halaman Upload File

Gambar 3.8 memperlihatkan halaman antarmuka yang digunakan untuk mengunggah file ke dalam sistem. Dalam tampilan ini, pengguna diminta untuk mengisi beberapa kolom informasi, seperti nama file yang akan diunggah, kategori dokumen untuk memudahkan pengelompokan, serta kolom sandi yang bersifat opsional guna memberikan perlindungan tambahan terhadap file tersebut. Setelah seluruh informasi terisi, pengguna dapat menekan tombol *Save Dokumen* yang berfungsi untuk menyimpan data file secara permanen ke dalam basis data sistem. Proses ini dirancang agar pengguna dapat melakukan unggahan file dengan mudah, aman, dan terstruktur.

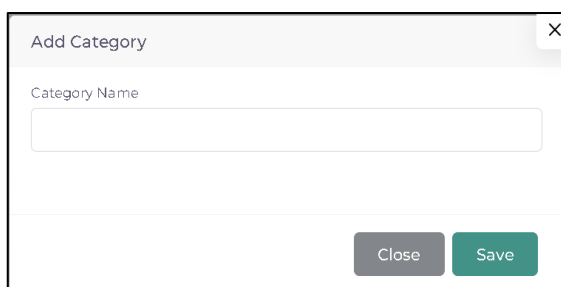
3.11. Halaman Kategori



Gambar 3.9 Halaman Kategori

Gambar 3.9 menampilkan antarmuka menu kategori, di mana pengguna dapat melihat daftar kategori yang telah tersimpan dalam sistem. Setiap kategori ditampilkan secara sistematis untuk mempermudah pengelompokan data atau dokumen berdasarkan jenis atau fungsinya. Menu ini berfungsi sebagai pusat pengelolaan kategori, memungkinkan pengguna untuk menambahkan, mengedit, atau menghapus kategori sesuai kebutuhan. Kehadiran fitur ini memberikan kemudahan dalam mengatur struktur data agar lebih rapi, terorganisir, dan efisien untuk diakses.

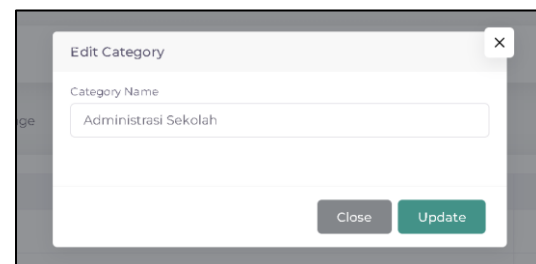
3.12. Halaman Tambah Kategori



Gambar 3.10 Halaman Tambah Kategori

Gambar 3.10 memperlihatkan tampilan antarmuka yang digunakan untuk menambahkan kategori baru ke dalam sistem. Pada halaman ini, tersedia sebuah field input untuk mengisi nama kategori yang akan dibuat. Setelah pengguna memasukkan nama kategori yang diinginkan, tombol *Save* dapat ditekan untuk menyimpan data tersebut ke dalam database sistem. Fitur ini dirancang agar pengguna dapat dengan mudah menyesuaikan pengelompokan data atau dokumen sesuai dengan struktur dan kebutuhan yang diinginkan, sehingga pengelolaan informasi menjadi lebih teratur dan efisien.

3.13. Halaman Edit Kategori



Gambar 3.11 Halaman Edit Kategori

Gambar 3.11 menampilkan antarmuka untuk mengubah data kategori yang sudah ada di dalam sistem. Dalam tampilan ini, pengguna disediakan sebuah field untuk mengedit nama kategori sesuai kebutuhan. Setelah melakukan perubahan, pengguna dapat menekan tombol *Save* untuk memperbarui dan menyimpan data kategori tersebut ke dalam sistem. Fitur ini memungkinkan proses pengelolaan kategori menjadi lebih fleksibel, sehingga struktur pengelompokan data dapat terus disesuaikan tanpa harus membuat ulang kategori dari awal.

3.14. Hasil Enkripsi Dokumen Pdf



Gambar 3.12 Hasil Enkripsi Dokumen Pdf

Gambar 3.12 memperlihatkan tampilan hasil proses enkripsi yang diterapkan pada sebuah file dokumen dengan ekstensi PDF. Pada gambar ini, dapat dilihat bagaimana file tersebut telah melalui mekanisme pengamanan sehingga isi dokumen menjadi terlindungi dari akses yang tidak sah. Proses enkripsi ini memastikan bahwa hanya pihak yang memiliki kunci atau izin yang dapat membuka dan membaca isi file PDF tersebut, sehingga meningkatkan tingkat keamanan data yang tersimpan di dalam sistem. Tampilan ini juga berfungsi sebagai bukti visual bahwa sistem berhasil melakukan proteksi terhadap file dokumen penting dengan efektif.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan implementasi yang dilakukan, sistem pengamanan dokumen yang dikembangkan mampu memenuhi kebutuhan pengguna dalam menyimpan dan mengelola file secara lebih efisien dan aman. Sistem ini mendukung multi-user, yaitu admin (kepala sekolah dan staf IT) serta guru. Fitur utama mencakup penyimpanan file berdasarkan folder, di mana file tersebut dapat dienkripsi untuk menjaga keamanan dokumen penting di lingkungan sekolah. Penggunaan Laravel terbukti efektif dalam meningkatkan efisiensi pengelolaan dan penyimpanan dokumen, sekaligus menjaga keamanan dan kerahasiaannya. Hasil penelitian

menunjukkan bahwa sistem berjalan sesuai dengan perancangan dan memenuhi kebutuhan pengguna.

Pengembangan sistem ke depan dapat mencakup penambahan fitur pendukung seperti notifikasi, pencarian dokumen, dan peningkatan antarmuka pengguna agar sistem lebih mudah digunakan. Selain itu, sistem juga dapat diintegrasikan dengan modul manajemen dokumen yang lebih lengkap untuk mendukung pengelolaan arsip secara menyeluruh dan aman.

DAFTAR PUSTAKA

- [1] ISO 27001, *Information Security Management System*. Switzerland. 2013.
- [2] APJII, "Survei Penetrasi Internet Indonesia 2024." Accessed: Feb. 18, 2025. [Online]. Available: <https://apjii.or.id>
- [3] NIST, *Cybersecurity Framework 2.0: Resource & Overview Guide*. 2018.
- [4] K. Philip and L. K. Kevin, *Marketing Management*, 15th ed. Pearson Education, 2016.
- [5] D. Hulu, B. Nadeak, and S. Aripin, "Implementasi Algoritma AES (Advanced Encryption Standard) Untuk Keamanan File Hasil Radiologi di RSUD Imelda Medan," vol. 4, no. 1, (2020)
- [6] S. William, *Cryptography and Network Security: Principles and Practice*. Boston: Pearson Education, 2017.
- [7] V. A. Kurniyanti and D. Murdiani, "Perbandingan Model Waterfall Dengan Prototype Pada Pengembangan System Informasi Berbasis Website," *Jurnal Syntax Fusion*, vol. 2, no. 8, pp. 669–675, Aug. 2022)
- [8] D. Kung, *Object-Oriented Software Engineering: An Agile Unified Methodology*. New York City: McGraw Hill, 2013.
- [9] R. Pressman and B. Maxim, *Software Engineering: A Practitioner's Approach*,

9th Edition. McGraw-Hill Education, 2020.

- [10] A. Dennis, H. B. Wixom, and R. M. Roth, *Systems Analysis and Design*. New York: Wiley, 2018.
- [11] L. Welling and L. Thomson, *PHP and MySQL Web Development*. Sams Publishing, 2003.
- [12] Laravel, "Laravel Documentation." Accessed: Feb. 25, 2025. [Online]. Available: <https://laravel.com/docs>
- [13] B. Beizer, *Software Testing Techniques*. Wiley India, 2008.
- [14] R. Mall, *Fundamentals Of Software Engineering, 5Th Ed*, 5th ed. PHI, 2018.
- [15] IEEE, *Software engineering - Software life cycle processes - Maintenance*. Switzerland: Institute of Electrical and Electronics Engineers, 2022.