Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

Rancang Bangun Password Manager Berbasis Web Menggunakan Framework Django Dan Enkripsi Fernet

Widarmawati Waruwu[1*], Yusnia Budiarti[2], Mutiara Nazwah[3], Angga Wibowo Saputro[4] Diwan Mardianus Laia[5]
[1,2,3,4,5] Program Studi Teknologi Informasi, Universitas Bina Sarana Informatika
*widarwaruwu116@gmail.com

Abstrak

Mayoritas penelitian password manager terdahulu masih berfokus pada enkripsi dasar tanpa derivasi kunci yang kuat dan belum mengimplementasikan arsitektur zero-knowledge secara konsisten, sehingga penyedia layanan atau pihak ketiga berpotensi mengakses data terenkripsi pengguna. Penelitian ini berkontribusi dengan merancang dan membangun aplikasi password manager berbasis web yang mengintegrasikan tiga lapisan mekanisme keamanan secara terpadu dalam satu ekosistem Django: derivasi kunci PBKDF2 dengan 100.000 iterasi SHA-256, salt acak unik per pengguna berbasis perangkat keras (os.urandom), dan enkripsi terautentikasi Fernet (AES-128-CBC + HMAC-SHA256). Kebaruan (novelty) penelitian ini terletak pada penerapan prinsip zero-knowledge secara ketat, di mana master password sama sekali tidak pernah disimpan di basis data, sehingga administrator sistem pun tidak dapat mengakses kredensial pengguna. Kombinasi salt personal dengan PBKDF2 juga menjamin bahwa dua pengguna dengan master password identik tetap menghasilkan kunci enkripsi yang berbeda, sesuatu yang belum ditemukan pada implementasi sejenis berbasis Django yang ada saat ini. Pengujian menggunakan metode Black Box Testing pada sepuluh skenario fungsional dan keamanan menunjukkan tingkat keberhasilan 100%, mencakup operasi CRUD, penolakan akses tidak sah, deteksi manipulasi ciphertext (tampering), serta verifikasi keunikan salt antar pengguna. Hasil penelitian membuktikan bahwa integrasi PBKDF2, salt unik, dan Fernet dalam framework Django menghasilkan sistem penyimpanan kredensial yang transparan, aman, dan tahan terhadap serangan brute force, rainbow table, serta manipulasi data basis data.

Kata kunci : Password Manager, Django, Enkripsi Fernet, PBKDF2, Zero-Knowledge, Keamanan Web.

Abstract

Most prior password manager studies rely on basic encryption without strong key derivation and have not consistently implemented a zero-knowledge architecture, leaving encrypted user data potentially accessible to service providers or third parties. This research contributes by designing and building a web-based password manager that integrates three security layers within a unified Django ecosystem: PBKDF2 key derivation with 100,000 SHA-256 iterations, hardware-based unique random salt per user (os.urandom), and authenticated Fernet encryption (AES-128-CBC + HMAC-SHA256). The novelty of this research lies in the strict enforcement of the zero-knowledge principle, where the master password is never stored in the database, making credentials inaccessible even to system administrators. The combination of personalized salt and PBKDF2 also ensures that two users sharing an identical master password always produce distinct encryption keys — a feature not found in existing Django-based implementations. Black Box Testing across ten functional and security scenarios yielded a 100% success rate, covering CRUD operations, unauthorized access rejection, ciphertext tampering detection, and cross-user salt uniqueness verification. The results demonstrate that integrating PBKDF2, unique salt, and Fernet within the Django framework produces a transparent, secure credential storage system resilient to brute force attacks, rainbow table attacks, and database-level data manipulation.

Keywords : Password Manager, Django, Fernet Encryption, PBKDF2, Zero-Knowledge, Web Security

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

1. Pendahuluan

Ketergantungan aktivitas masyarakat modern pada akun digital memicu tantangan baru dalam pengelolaan autentikasi kata sandi[1]. Banyaknya jumlah akun membuat pengguna cenderung menggunakan kata sandi yang sederhana atau menerapkan praktik password reuse (satu kata sandi untuk berbagai layanan). Praktik ini sangat rentan terhadap serangan keamanan seperti credential stuffing dan brute force attack, sebab kebocoran pada satu akun dapat mengekspos akun lainnya. Contoh riil dari dampak fatal risiko ini adalah insiden pelanggaran data 23andMe pada tahun 2023, di mana eksploitasi kata sandi berhasil memberikan akses tidak sah ke ribuan akun dan berdampak luas pada data yang saling terhubung[2].

Untuk memitigasi risiko tersebut, aplikasi password manager dikembangkan sebagai solusi penyimpanan terpusat agar pengguna dapat menerapkan kata sandi yang unik dan kuat di setiap akun[3]. Kendati demikian, sentralisasi ini menghadirkan tantangan baru karena basis data tunggal tersebut berpotensi menjadi single point of failure (titik serangan utama). Jika mekanisme perlindungan data pada aplikasi tidak dirancang dengan matang, insiden kebocoran basis data justru akan menyebabkan seluruh informasi kredensial pengguna terekspos secara massal[4]. Meskipun beberapa penelitian terdahulu telah menerapkan teknik enkripsi pada password manager, mayoritas masih berfokus pada enkripsi dasar tanpa derivasi kunci yang kuat serta belum mengoptimalkan pendekatan zero-knowledge. Celah tersebut membuka peluang bagi penyedia layanan atau pihak ketiga untuk mengakses data terenkripsi, yang diperparah oleh penggunaan kunci enkripsi non-personalisasi[5]. penelitian ini bertujuan merancang sistem password manager berbasis web menggunakan framework Django dengan keamanan berlapis. Melalui kombinasi algoritma Fernet untuk enkripsi dan autentikasi integritas data, serta metode PBKDF2 dengan unique salt untuk personalisasi kunci, sistem ini diharapkan mampu menegakkan prinsip zero-knowledge secara kokoh, sehingga informasi kata sandi tetap tidak dapat dibaca tanpa master password pengguna meskipun basis data berhasil ditembus[6].

2. Tinjauan Pustaka

2.1. Penelitian Terkait

Penelitian terkait digunakan sebagai acuan dasar dan pembanding untuk mempertegas posisi serta keaslian riset ini dalam lingkup keamanan informasi dan pengelolaan kredensial digital. Beberapa kajian terdahulu yang menjadi landasan utama dalam penelitian ini dijabarkan sebagai berikut :

1. Aspek Usabilitas Antarmuka : Studi longitudinal membuktikan kemudahan penggunaan (usability) antarmuka web sangat

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

krusial bagi pengguna awam untuk mencegah pengulangan kata sandi (password reuse). Riset ini mengadopsi konsep tersebut melalui perancangan antarmuka adaptif berbasis Bootstrap[1].

2. Isolasi Kredensial Arsitektur Web: Penelitian menekankan pentingnya isolasi kredensial dan validasi token untuk keamanan autentikasi web. Sistem ini menerapkannya dengan tidak pernah mengirimkan atau menyimpan master password ke dalam basis data[2].

3. Mitigasi Kerentanan Sisi Klien: Analisis keamanan platform pengelola kata sandi mengidentifikasi kerentanan skrip browser akibat kesalahan logika dan otorisasi yang dapat mengekstrak kredensial. Temuan ini mendasari sistem untuk memperketat isolasi data client-side menggunakan Django dan Fernet[3].

4. Penguatan Derivasi Kunci: Evaluasi fungsi derivasi kunci membuktikan peningkatan iterasi hash model dinamis PBKDF2 signifikan memperlambat komputasi brute force peretas. Sistem mengimplementasikan temuan ini lewat penerapan PBKDF2 sebanyak 100.000 iterasi dengan SHA-256[4].

5. Pencegahan Serangan Rainbow Table: Analisis komparasi algoritma hash dan teknik salting menunjukkan salt acak unik per pengguna efektif mematahkan serangan rainbow table. Penelitian ini mengadopsi fungsi tersebut melalui pembentukan salt acak berbasis perangkat keras (os.urandom)[5]

2.2. Landasan Teori

1. Password Manager

Password Manager adalah perangkat lunak yang dirancang untuk menyimpan, mengelola, dan mengorganisasikan kredensial akun digital secara terpusat. Guna menjamin keamanan mutlak pada platform berbasis web, sistem harus mengadopsi arsitektur zero-knowledge. Dalam arsitektur ini, penyedia layanan atau administrator server sama sekali tidak memiliki pengetahuan teknis terhadap data sensitif plaintext milik pengguna. Seluruh proses kriptografi penentu kunci dilakukan menggunakan parameter yang hanya diketahui oleh pengguna (master password), sehingga data yang disimpan di dalam basis data sudah berupa ciphertext.[11]

2. Framework Django

Django merupakan framework pengembangan aplikasi web berbasis bahasa pemrograman Python yang mengusung filosofi security by default. Django menyediakan lapisan abstraksi tingkat tinggi serta modul keamanan bawaan terintegrasi, seperti perlindungan otomatis terhadap Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), dan SQL Injection melalui penggunaan Object-Relational Mapping (ORM). Struktur komponen Django yang teratur memudahkan pemisahan antara logika pemrosesan kriptografi (backend) dan penyajian

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

antarmuka halaman (frontend) [8]

3. Fungsi Derivasi Kunci PBKDF2

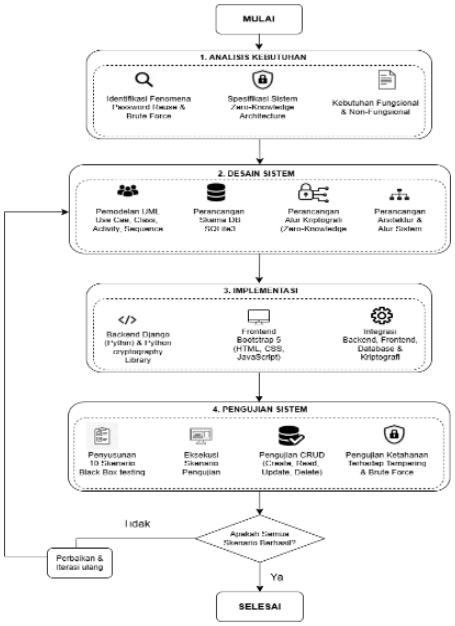
Password-Based Key Derivation Function 2 (PBKDF2) adalah standar kriptografi (RFC 2898) yang digunakan untuk meregangkan kata sandi (key stretching) menjadi kunci kriptografi yang kuat. Algoritma ini bekerja dengan menerapkan fungsi pseudorandom (seperti HMAC-SHA256) ke dalam kata sandi utama bersamaan dengan salt acak secara berulang-ulang (iteration).[12] Jumlah iterasi yang tinggi (misalnya 100.000 kali) bertujuan memperlambat kecepatan eksekusi komputasi secara sengaja, sehingga meningkatkan resistensi kunci terhadap serangan tebakan massal seperti brute-force maupun dictionary attacks.

4. Enkripsi terautentikasi Fernet

Fernet adalah spesifikasi skema enkripsi simetris yang menyediakan jaminan bahwa data yang dienkripsi tidak dapat dibaca atau dimodifikasi tanpa kunci yang sah. Di bawah kapnya, Fernet memanfaatkan algoritma standar industri AES (Advanced Encryption Standard) dengan panjang kunci 128-bit dalam mode operasi CBC (Cipher Block Chaining). Keunggulan utama Fernet adalah penerapan Authenticated Encryption menggunakan HMAC-SHA256. Setiap token Fernet terdiri dari komponen penanda waktu (timestamp), Initialization Vector (IV), ciphertext, dan nilai tanda tangan digital HMAC. Jika ciphertext dimanipulasi di basis data, verifikasi HMAC akan gagal dan memicu eksepsi keamanan, menggagalkan proses dekripsi demi menjaga integritas data.

2.3. Tahapan Penelitian

Penelitian ini dilaksanakan secara sistematis mengikuti kerangka kerja sekuensial berdasarkan metodologi Waterfall. Alur tahapan perancangan hingga pengujian sistem digambarkan secara rinci pada diagram berikut:



Gambar 1 Diagram Alur Penelitian

1. Analisis Kebutuhan: Mengidentifikasi akar permasalahan keamanan kredensial, seperti kebocoran data akibat kebiasaan pengelolaan kata sandi yang buruk. Berdasarkan analisis tersebut, dirumuskan spesifikasi teknis kebutuhan fungsional dan keamanan sistem pengelola kata sandi

2. Desain Sistem: Menerjemahkan kebutuhan ke

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

dalam bentuk rancangan arsitektur. Tahap ini menghasilkan diagram UML (seperti Use Case dan Sequence Diagram) untuk memetakan logika pemisahan kunci dan hak akses data terenkripsi.

3. Implementasi: Mentransformasikan seluruh rancangan desain ke dalam kode program riil menggunakan bahasa pemrograman Python. Kriptografi Fernet dan derivasi kunci PBKDF2 diintegrasikan ke dalam ekosistem aplikasi web Django.

4. Pengujian: Melakukan verifikasi menyeluruh terhadap fungsionalitas aplikasi dan efektivitas algoritma enkripsi menggunakan metode Black Box Testing untuk menjamin seluruh komponen bekerja andal sesuai standar spesifikasi.

3. Metode Penelitian

3.1. Metode Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui tiga teknik utama, yaitu:

1. Studi Literatur, dengan mengumpulkan dan mempelajari referensi jurnal ilmiah, dokumentasi Fernet, PBKDF2, dan zero-knowledge architecture.
2. Observasi Sistem Existing, dilakukan terhadap beberapa aplikasi password manager populer guna mengidentifikasi fitur keamanan, mekanisme penyimpanan data, serta keterbatasan yang ada.
3. Analisis Kebutuhan, untuk merumuskan spesifikasi fungsional dan non-fungsional sistem berdasarkan data yang diperoleh.

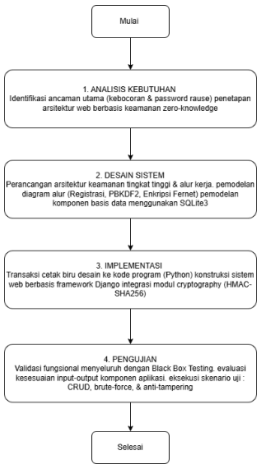
3.2. Prosedur Pengembangan Sistem

Pengembangan password manager ini menerapkan metode Waterfall sekuensial untuk memastikan seluruh spesifikasi fungsional dan arsitektur keamanan berlapis dirancang secara matang sebelum tahap implementasi (Gambar 2). Prosedur ini diawali dengan analisis kebutuhan untuk mengidentifikasi ancaman keamanan seperti password reuse guna menetapkan spesifikasi sistem berbasis zero-knowledge. Selanjutnya, fase desain menerjemahkan kebutuhan tersebut menjadi cetak biru arsitektur, pemodelan basis data SQLite3, dan alur enkripsi-dekripsi. Cetak biru ini kemudian dieksekusi pada tahap implementasi dengan mentransformasikan rancangan menjadi kode program Python menggunakan framework Django dan pustaka cryptography untuk menjamin integritas data lewat HMAC-SHA256. Akhirnya, tahap pengujian dilakukan sebagai fase akhir untuk memverifikasi fungsionalitas dan keandalan sistem melalui Black Box Testing yang mencakup evaluasi operasi CRUD, ketahanan terhadap serangan brute force, serta deteksi manipulasi data (tampering).[6]

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

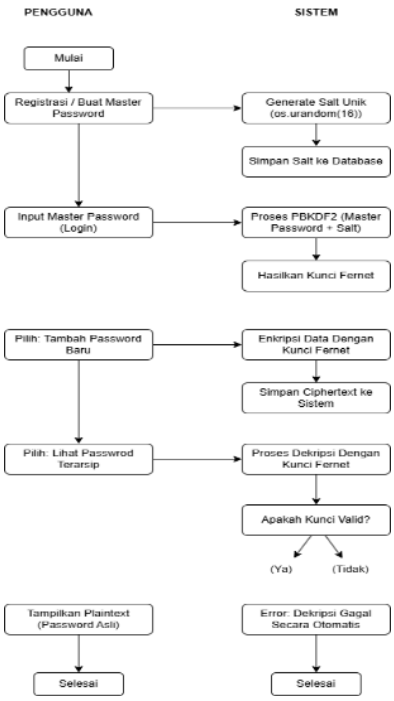
DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065



Gambar 2. Bagan Alur Prosedur Pengembangan Sistem

3.3. Arsitektur Keamanan dan Alur Sistem

Sistem menerapkan keamanan berlapis berbasis zero-knowledge architecture, di mana master password tidak pernah disimpan dalam basis data[6]. Alur kerja sistem dirancang sebagai berikut (Gambar 3):



Gambar 3 Diagram Alur Sistem

1. Registrasi: Sistem membangkitkan salt acak unik berbasis perangkat keras menggunakan fungsi os.urandom(16) yang disimpan di basis data bersama data pengguna.

2. Login & Derivasi Kunci: Master password dan salt diproses menggunakan metode PBKDF2 melalui penguatan 100.000 iterasi SHA-256 untuk menderivasi kunci enkripsi Fernet 32-byte[15].

3. Enkripsi & Dekripsi: Kunci Fernet (AES-128-CBC + HMAC-SHA256) digunakan untuk mengamankan data kredensial baru menjadi ciphertext. Saat data diakses, proses dekripsi dilakukan menggunakan kunci yang sama. Jika master password salah atau ciphertext di basis data dimanipulasi, proses dekripsi otomatis ditolak oleh sistem[16].

Kunci Fernet ini menjadi instrumen tunggal untuk mengenkripsi kata sandi baru menjadi ciphertext di basis data, sekaligus mendekripsinya kembali menjadi teks asli saat diakses pengguna. Jika pengguna memasukkan master password yang salah saat login, PBKDF2 akan menghasilkan kunci yang berbeda sehingga proses dekripsi Fernet otomatis ditolak dan gagal.

3.4. Mode Pengujian

Pengujian dilakukan menggunakan metode Black Box Testing untuk mengevaluasi aspek fungsional dan keamanan sistem tanpa memeriksa kode internal. Pengujian fungsional mencakup validasi operasi CRUD (Create, Read,

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

Update, Delete) pada kredensial pengguna. Sementara pengujian keamanan dilakukan melalui tiga skenario: simulasi kegagalan login dengan kata sandi salah, modifikasi ciphertext secara manual di basis data untuk menguji fitur anti-tampering Fernet, serta verifikasi keunikan kunci enkripsi pada dua pengguna dengan master password yang sama

3.5. Lokasi Penelitian

Penelitian ini dilaksanakan secara mandiri pada lingkungan pengembangan perangkat lunak (development environment) menggunakan perangkat komputer pribadi secara lokal dengan memanfaatkan framework Django, basis data SQLite3, dan bahasa pemrograman Python. Waktu pelaksanaan riset mencakup sepanjang periode penyusunan tugas akhir pada Program Studi Teknologi Informasi, Universitas Bina Sarana Informatika Eksperimen ini menerapkan pendekatan rekayasa perangkat lunak (software engineering) dengan mengadopsi metode pengembangan sistem Waterfall. Fokus utama penelitian terarah pada rancang bangun aplikasi password manager berbasis web yang mengintegrasikan mekanisme enkripsi untuk meningkatkan keamanan penyimpanan kredensial pengguna. Sistem dirancang berdasarkan prinsip zero-knowledge, sehingga data kata sandi tetap tidak dapat dibaca meskipun basis data berhasil diakses tanpa otorisasi[11].

4. Hasil dan Pembahasan

4.1. Hasil Penelitian

1. Implementasi Antarmuka Sistem

Penelitian ini berhasil mengimplementasikan aplikasi password manager berbasis web menggunakan framework Django dengan antarmuka responsif berbasis Bootstrap. Sistem memenuhi seluruh kebutuhan fungsional meliputi registrasi, autentikasi, hingga manajemen kredensial (CRUD). Halaman login mengendalikan keamanan akses dengan mewajibkan penggunaan master password sebagai kunci utama (Gambar 4). Setelah autentikasi berhasil, pengguna dapat mengelola daftar kata sandi pada halaman utama (Gambar 5), di mana backend memproses seluruh interaksi data secara terenkripsi otomatis. [17]

Gambar 4 Tampilan Halaman Login Aplikasi Password Manager

Gambar 5 Tampilan Halaman Utama

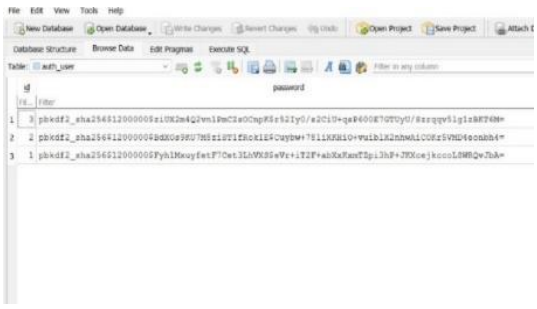
Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

2. Keamanan Data di Dalam Database

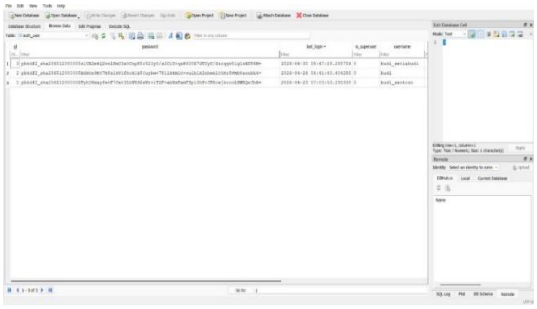
Salah satu hasil implementasi terpenting dalam sistem ini adalah cara penyimpanan data di dalam basis data SQLite3. Berbeda dengan sistem yang menyimpan kata sandi dalam bentuk teks biasa (plaintext), sistem ini menyimpan kata sandi pengguna dalam kolom encrypted password yang berisi rangkaian karakter acak (ciphertext) hasil enkripsi Fernet. Karakter-karakter acak tersebut tidak dapat dibaca atau diinterpretasikan tanpa kunci dekripsi yang benar, sehingga meskipun seseorang berhasil membuka file database secara langsung alat seperti DB Browser for SQLite, isi kata sandi tetap tidak dapat diketahui. Contoh ciphertext yang dihasilkan oleh Fernet tampak seperti: gAAAAABj5X2K..... (string Base64 URL-safe sepanjang 88 hingga 200+ karakter tergantung panjang plaintext). Fernet ini merupakan representasi dari IV (16 byte), ciphertext (terpadding ke kelipatan 16 byte). dan HMAC (32 byte) yang digabungkan dan di-encode menggunakan Base64 URL-safe



Gambar 6 Tampilan Isi Database SQLite Memperlihatkan Kolom encrypted_password Berisis Ciphertext

3. Analisis Representasi Data Terenkripsi

Validasi pada basis data SQLite3 menunjukkan sistem menyimpan seluruh informasi sensitif dalam format ciphertext (Gambar 7). Hal ini membuktikan keberhasilan rancangan zero-knowledge architecture; pihak tidak berwenang tetap tidak dapat membaca informasi asli meskipun berhasil mengakses file basis data. Algoritma Fernet menjamin data tersimpan tidak hanya terenkripsi tetapi juga terautentikasi, sehingga menjaga integritas data dari upaya manipulasi manual pada level basis data.



Gambar 7 Tampilan Data Terenkripsi pada Tabel Database SQLite

3. Evaluasi Fungsional dan Keamanan (Black Box Testing).

Rekapitulasi pada Tabel 1 menunjukkan tingkat keberhasilan 100% untuk seluruh fitur utama sistem. Selain menyukseskan fungsi CRUD, sistem terbukti tangguh menolak master password yang salah dan mendeteksi perubahan data (tampering). Pengujian juga mengonfirmasi bahwa penggunaan unique salt berhasil menghasilkan kunci enkripsi yang berbeda untuk setiap pengguna, meskipun mereka

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

mendaftarkan kata sandi yang identik. [18] .

Tabel 1 Rekapitulasi Hasil Pengujian Fungsi dan Keamanan Sistem

No	Fungsi yang Diuji	Kondisi Pengujian	Hasil yang Diharapkan	Status
1	Registrasi Akun	Input data pengguna baru dengan master password valid	Akun berhasil dibuat, salt unik terbentuk, dan pengguna diarahkan ke halaman login	Berhasil
2	Login Pengguna	Input username dan master password yang benar	Pengguna berhasil masuk dan sesi aktif	Berhasil
3	Login Gagal	Input master password yang salah	Sistem menolak akses dan menampilkan pesan error	Berhasil
4	Tambah Kata Sandi	Input nama akun, username, dan kata sandi baru	Data tersimpan dalam bentuk ciphertext di database	Berhasil
5	Lihat Kata Sandi	Klik tombol dekripsi pada entri kata sandi	Kata sandi asli ditampilkan hanya jika master password aktif benar	Berhasil
6	Edit Kata Sandi	Ubah kata sandi yang sudah ada melalui halaman edit	Data diperbarui dan disimpan ulang dalam bentuk ciphertext baru	Berhasil
7	Hapus Kata Sandi	Hapus entri kata sandi yang dipilih	Entri dihapus dari database dan tidak muncul di daftar	Berhasil
8	Keamanan Fernet	Modifikasi manual ciphertext di database kemudian coba dekripsi	Sistem menolak dekripsi dan menampilkan pesan InvalidToken	Berhasil
9	Salt Unik per Pengguna	Dua pengguna mendaftar dengan master password identik	Kunci enkripsi yang dihasilkan berbeda karena salt unik	Berhasil
10	PBKDF2 Iterasi	Verifikasi jumlah iterasi PBKDF2 pada kode sumber	Terkonfirmasi 100.000 iterasi menggunakan SHA-256	Berhasil

4.2. Pembahasan

1. Studi Komparatif

Integrasi metode PBKDF2 dan algoritma Fernet dalam penelitian ini membentuk mekanisme keamanan berlapis yang andal untuk mengamankan data kredensial pengguna.[19] Keunggulan sistem ini dibanding arsitektur konvensional adalah penerapan unique salt personal bagi setiap pengguna. [20]

Implementasi nilai acak yang berbeda ini memastikan kunci enkripsi selalu unik, meskipun beberapa pengguna mendaftarkan master password yang identik. Karakteristik tersebut mengeliminasi risiko serangan yang memanfaatkan pola hash seragam pada basis data, sekaligus membedakan sistem ini dari penelitian terdahulu. [14].

2. Analisis Ketahanan Terhadap Serangan

Sistem menguatkan keamanan level komputasi melalui penerapan fungsi derivasi kunci PBKDF2

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

dengan standar 100.000 iterasi. Konfigurasi ini memberikan beban kompleksitas tambahan yang memaksa setiap percobaan autentikasi melewati siklus kalkulasi yang sama. Dampaknya, kebutuhan sumber daya dan waktu komputasi penyerang untuk mengeksekusi skenario brute force meningkat drastis. Selain tangguh terhadap brute force, sistem memberikan proteksi tinggi terhadap ancaman manipulasi data (tampering). Hasil pengujian teknis membuktikan setiap upaya perubahan ciphertext ilegal pada basis data langsung memicu kegagalan proses dekripsi. Mekanisme penolakan otomatis ini mengonfirmasi efektivitas fitur authenticated encryption milik Fernet dalam memverifikasi dan menjaga integritas data kredensial dari intervensi pihak ketiga

3. Perbandingan dengan Penelitian Terkait

Karakteristik keamanan sistem yang dikembangkan sebanding dengan standar password manager modern pada aspek derivasi kunci, implementasi salt unik, dan penegakan prinsip zero-knowledge (Tabel 2). Penelitian ini berhasil mengonstruksikan teori dari kajian Cabarcos dkk. mengenai urgensi perlindungan kredensial lewat enkripsi kuat, serta studi Alqahtani tentang relevansi kecukupan jumlah iterasi PBKDF2 ke dalam sistem riil. Kontribusi utama penelitian ini terletak pada visualisasi konsep keamanan tersebut menjadi aplikasi berbasis web siap pakai memanfaatkan keandalan ekosistem framework Django.

Tabel 2. Perbandingan Sistem dengan Pendekatan Lain

Aspek	Sistem Ini (Django+Fernet)	Password Manager Umum	Penyimpanan Plaintext
Algoritma Enkripsi	AES-128-CBC (Fernet)	AES-256 / ChaCha20	Tidak ada
Autentikasi Data	HMAC-SHA256	Tergantung implementasi	Tidak ada
Derivasi Kunci	PBKDF2 (100.000 iter.)	PBKDF2 / Argon2	Tidak ada
Salt per Pengguna	Ya, unik (os.urandom)	Ya	Tidak relevan
Zero-Knowledge	Ya	Sebagian besar Ya	Tidak
Pemulihan Password	Tidak (by design)	Tergantung layanan	Ya (risiko tinggi)
Open Source	Ya (Django + Python)	Sebagian	Bervariasi

4.3 Kelebihan dan Batasan Penelitian

Evaluasi mendalam menunjukkan keunggulan utama sistem pada penegakan prinsip zero-knowledge, personalisasi unique salt berbasis perangkat keras, serta kapabilitas authenticated encryption Fernet dalam mendeteksi tampering. Arsitektur bawaan framework Django memperkuat keamanan ini melalui proteksi security by default terhadap penetrasi siber berbasis web seperti SQL Injection dan Cross-Site Request Forgery (CSRF).

Kendati lulus pengujian fungsional dengan tingkat keberhasilan 100%, sistem memiliki beberapa batasan teknis untuk peluang pengembangan ke depan:

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

1. Skalabilitas Basis Data: Penggunaan SQLite3 membatasi performa pada skala produksi makro akibat kerentanan terhadap operasi tulis bersamaan (concurrent write).
2. Karakteristik Derivasi Kunci: Algoritma PBKDF2 belum memiliki sifat memory-hard, sehingga secara teoritis lebih rentan terhadap serangan paralel berbasis perangkat keras (GPU/ASIC) dibanding algoritma modern seperti Argon2id.
3. Konsekuensi Kebijakan Keamanan: Kepatuhan mutlak pada prinsip zero-knowledge meniadakan fitur pemulihan akun (account recovery), sehingga kelalaian pada master password akan menghilangkan data kredensial secara permanen.
4. Cakupan Pengujian: Evaluasi keamanan masih terbatas pada performa fungsional backend melalui Black Box Testing, serta belum mencakup analisis penetrasi mendalam pada jaringan publik, skrip sisi klien (client-side), maupun simulasi serangan intersep seperti Man-in-the-Middle (MitM).

5. Kesimpulan

Penelitian ini berhasil merancang dan membangun sistem password manager berbasis web menggunakan framework Django dan enkripsi Fernet yang menerapkan arsitektur keamanan berlapis berbasis prinsip zero-knowledge. Seluruh lapisan keamanan yang direncanakan berhasil diimplementasikan dengan baik, mencakup antarmuka responsif berbasis Bootstrap 5, penyimpanan terenkripsi algoritma Fernet (AES-128-CBC + HMAC-SHA256), serta derivasi kunci PBKDF2 dengan 100.000 iterasi dan salt unik per pengguna. Kepatuhan terhadap prinsip zero-knowledge terbukti efektif karena master password sama sekali tidak pernah disimpan dalam database, sehingga tidak ada pihak termasuk administrator yang dapat mengakses kata sandi pengguna. Melalui Black Box Testing, sepuluh skenario uji yang meliputi fungsi CRUD, keamanan autentikasi, ketahanan brute-force, deteksi manipulasi ciphertext (tampering), dan verifikasi salt unik seluruhnya dinyatakan berhasil sesuai spesifikasi. Terakhir, perbandingan dengan pendekatan lain menunjukkan bahwa meski sistem ini menggunakan AES-128 (yang fasenya berada di bawah AES-256 pada aplikasi komersial namun tetap aman menurut standar NIST), sistem ini unggul dalam hal transparansi implementasi (open source), konsistensi prinsip zero-knowledge, serta integrasi enkripsi terautentikasi yang mampu mendeteksi manipulasi data secara otomatis

6. Daftar Pustaka

[1] N. Weckwerth, B. Xia, and J. Zhang, “Password Manager Security,” pp. 1–11, 2020.
[2] R. Holthouse, S. Owens, and S. Bhunia, “The 23andMe Data Breach : Analyzing Credential Stuffing Attacks , Security

Infotek : Jurnal Informatika dan Teknologi
Vol. 9 No. 2, Juli 2026
Hal.537-548

e-ISSN 2614-8773

DOI : 10.29408/jit.v9i2.35065 Link : https://dx.doi.org/10.29408/jit.v9i2.35065

Vulnerabilities , and Mitigation Strategies".

[3] A. Cherry, K. Barmpis, and S. F. Shahandashti, "The Emperor is Now Clothed : A Secure Governance Framework for Web User Authentication through Password Managers ⋆," 2024.

[4] I. Contributors, "Cryptography Documentation," 2026.

[5] J. Kim, M. Song, M. Seo, Y. Jin, S. Shin, and J. Kim, "P ASS RE FINDER -FL : Privacy-Preserving Credential Stuffing Risk Prediction via Graph-Based Federated Learning for Representing Password Reuse between Websites ⋆".

[6] S. F. Manullang and J. Sembiring, "Implementasi Kriptografi Pengamanan Data File Dokumen Menggunakan Algoritma Advanced Encryption Standard Mode Chiper Block Chaining," pp. 87–95, 2001.

[7] C. Computing and S. Id, "Enhanced File Transfer Security in Django Web Applications with TOTP-Based Multi-Factor Authentication and Blowfish / AES Encryption on AWS Cloud".

[8] Z. Li, W. He, D. Akhawe, and D. Song, "The Emperor ' s New Password Manager : Security Analysis of Web-based Password Managers," 2014.

[9] A. A. S. Alqahtani, "Key Derivation : A Dynamic PBKDF2 Model for Modern Cryptographic Systems," 2025.

[10] A. Sha-, N. Sitorus, J. Sharon, G. Sinaga, and S. L. Samosir, "Analisis Kinerja Algoritma Hash pada Keamanan Data : Perbandingan," vol. 2, no. 2, 2024.

[11] P. A. Cabarcos, K. Security, and P. Mayer, "A Longitudinal Study on the Usability of Password Managers for Novice Users This paper is included in the Proceedings of the Twenty-First Symposium on Usable Privacy and Security .," 2025.

[12] A. A. S. Alqahtani, "Key Derivation : A Dynamic PBKDF2 Model for Modern Cryptographic Systems," 2025.

[13] M. Ridwan and A. R. Yusuf, "Framework Bootstrap Untuk Pengelolaan Data Akademik Dan Administrasi," vol. 2, no. 2, pp. 112–124, 2025.

[14] D. F. Sari, A. M. Sari, I. Janah, and J. S. Asri, "Analisis Komparasi Algoritma Hash (Md5 , SHA-2 , SHA-3 , SHA-512) Dan Teknik Salting Untuk Peningkatan Keamanan Data," vol. 6, no. 1, pp. 7379–7384, 2026.

[15] S. Salamatian, W. Huleihel, A. Beirami, A. Cohen, and M. Muriel, "Centralized vs Decentralized Targeted Brute-Force Attacks : Guessing with Side-Information," pp. 1–14, 2017.

[16] A. H. Jaya, A. Rosyidi, P. Studi, M. Informatika, and S. Indonesia, "Implementasi Aes-256 Dalam Aplikasi Manajemen Password Menggunakan Bahasa Pemrograman Java Berbasis," no. November 2024, pp. 1187–1196.

[17] N. F. Aprilia, D. Mafazi, A. R. Muchtar, K. Afif, A. Rohim, and N. Ilham, "Penerapan Algoritma AES untuk Enkripsi pada Halaman Register serta Penerapan AES untuk Deskripsi pada Halaman Login Website," vol. 1, no. April, pp. 75–82, 2023.

[18] E. Adventure, "Pengujian black box pada Website dengan Metode Robustness Testing," vol. 3, no. 2, pp. 93–96, 2022.

[19] S. Informatika, U. Majalengka, F. Recognation, and S. Keamanan, "4 1234," vol. 7, no. 1, pp. 205–215, 2024.

[20] R. Rahman, A. Yosua, and N. Leksona, "Serangan Man-In-The-Middle (MITM) di Jaringan Publik : Studi dan Solusi Simulasi Serangan Password Cracking Menggunakan Hydra," vol. 1, no. 4, pp. 2145–2156, 2025.