**Implementasi Super Enkripsi Caesar dan Scytale untuk Keamanan Database terhadap Serangan SQL Injection**

**Latar Belakang**

Ketergantungan sistem informasi modern pada database untuk menyimpan dan mengelola data sangatlah tinggi. Database telah menjadi jantung dari berbagai aplikasi, mulai dari situs web, aplikasi mobile, hingga sistem enterprise yang kompleks, yang setiap hari memproses jutaan gigabita data sensitif. Seiring dengan peran vitalnya, keamanan database menjadi aspek yang krusial. Kegagalan dalam mengamankan database dapat mengakibatkan kerugian finansial, kerusakan reputasi, dan pelanggaran privasi yang serius bagi individu maupun organisasi.

Ancaman utama yang secara konsisten menargetkan database adalah serangan SQL Injection. Serangan ini terjadi ketika pihak ketiga yang tidak berwenang menyisipkan perintah SQL berbahaya melalui kolom input pada sebuah aplikasi, dengan tujuan untuk memanipulasi atau mengambil alih database. Meskipun terlihat sederhana, dampak dari serangan ini bisa sangat merusak, mulai dari pencurian data (data exfiltration), modifikasi data yang tidak sah, hingga penghapusan seluruh tabel atau database.

Meskipun berbagai teknik pencegahan telah dikembangkan, seperti penggunaan prepared statements atau parameterized queries dan Web Application Firewall (WAF), banyak sistem, terutama aplikasi warisan (legacy system), masih memiliki celah kerentanan. Ketika lapisan pertahanan ini berhasil ditembus dan penyerang berhasil mengekstrak data, masalah yang lebih besar muncul: data tersebut sering kali tersimpan dalam format teks asli (plaintext). Hal ini menciptakan risiko kebocoran data masif yang langsung dapat dimanfaatkan oleh pihak penyerang.

Untuk mengatasi masalah tersebut, penelitian ini mengusulkan sebuah solusi yang berlandaskan pada prinsip pertahanan berlapis (defense-in-depth). Jika lapisan pertama (pencegahan) gagal, maka lapisan kedua harus mampu memitigasi kerusakan. Solusi yang diajukan adalah enkripsi data pada tingkat aplikasi. Dengan pendekatan ini, semua data sensitif dienkripsi terlebih dahulu sebelum disimpan ke dalam database. Dengan demikian, sekalipun seorang penyerang berhasil melakukan SQL Injection dan mencuri data, informasi yang mereka peroleh hanyalah sekumpulan sandi (ciphertext) yang tidak dapat dibaca dan tidak memiliki nilai.

Secara spesifik, metode yang akan diimplementasikan dalam penelitian ini adalah "Super Enkripsi", yaitu sebuah sistem enkripsi produk yang menggabungkan dua algoritma kriptografi klasik: Caesar Cipher dan Scytale Cipher. Caesar Cipher, sebagai sandi substitusi, akan mengubah setiap karakter dalam pesan, sementara Scytale Cipher, sebagai sandi transposisi, akan mengubah urutan karakter tersebut. Dengan menerapkan kedua sandi ini secara berurutan, diharapkan akan menghasilkan ciphertext yang lebih kompleks dibandingkan jika hanya menggunakan salah satu metode saja, namun tetap ringan secara komputasi.

Jika dibandingkan dengan penelitian terdahulu, pendekatan ini mengambil posisi yang unik. Sebagian besar penelitian modern fokus pada pencegahan serangan SQL Injection atau menggunakan enkripsi modern yang sangat kuat seperti AES untuk melindungi data. Sementara itu, penelitian lain yang membahas sandi klasik biasanya hanya berfokus pada analisis kelemahannya. Penelitian ini berbeda karena mencoba menjembatani ketiganya: menggunakan sandi klasik sebagai solusi mitigasi dampak untuk masalah keamanan modern.

Kebaruan (novelty) dari penelitian ini tidak terletak pada penemuan algoritma baru, melainkan pada penerapan dan analisis kinerja dari kombinasi unik Caesar Cipher dan Scytale Cipher sebagai lapisan keamanan database untuk melawan dampak pencurian data akibat SQL Injection. Tujuan utama dari penelitian ini adalah merancang dan mengimplementasikan sistem "Super Enkripsi" ini ; menganalisis dampaknya terhadap kinerja operasi database ; serta memvalidasi efektivitasnya dalam membuat data hasil curian menjadi tidak dapat dibaca.

Pada akhirnya, penelitian ini diharapkan dapat memberikan kontribusi signifikan baik secara teoretis maupun praktis. Secara teoretis, penelitian ini akan menyajikan data empiris mengenai kinerja dan kelayakan penggunaan kriptografi klasik sebagai lapisan keamanan tambahan. Secara praktis, penelitian ini menghasilkan sebuah model konseptual yang teruji untuk melindungi data sensitif, meningkatkan kesadaran akan pentingnya enkripsi data at-rest, dan menawarkan alternatif solusi keamanan yang ringan untuk aplikasi dengan kebutuhan spesifik.

**Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah diuraikan, maka permasalahan utama dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan algoritma "Super Enkripsi" yang menggabungkan Caesar Cipher dan Scytale Cipher untuk proses enkripsi dan dekripsi data pada aplikasi berbasis database?
2. Bagaimana dampak implementasi enkripsi ini terhadap kinerja (*overhead*) operasi database, khususnya pada waktu yang dibutuhkan untuk proses INSERT (penyimpanan data) dan SELECT (pengambilan data)?
3. Sejauh mana efektivitas "Super Enkripsi" dalam mengamankan konten data sehingga tidak dapat dibaca ketika berhasil diekstrak melalui simulasi serangan SQL Injection?

**Batasan Masalah**

Agar penelitian ini lebih terarah dan fokus, ditetapkan batasan-batasan masalah sebagai berikut:

1. Fokus Penelitian: Penelitian ini berfokus pada enkripsi data di tingkat aplikasi untuk melindungi kerahasiaan data di dalam database (*data at rest*), bukan pada metode pencegahan serangan SQL Injection itu sendiri (seperti WAF atau validasi input).
2. Metode Enkripsi: Metode kriptografi yang digunakan terbatas hanya pada kombinasi Caesar Cipher dan Scytale Cipher.
3. Lingkungan Pengembangan: Purwarupa sistem dibangun sebagai aplikasi web menggunakan bahasa pemrograman javascript dengan sistem manajemen database MySQL.
4. Lingkup Enkripsi: Enkripsi hanya diterapkan pada kolom-kolom tabel database yang dianggap sensitif (contoh: nama pengguna, email, kata sandi), bukan pada keseluruhan database.
5. Simulasi Serangan: Serangan SQL Injection yang disimulasikan terbatas pada teknik UNION-based yang bertujuan untuk ekstraksi atau pembacaan data.

**Tujuan dan Manfaat**

**Tujuan**

1. Mengimplementasikan algoritma "Super Enkripsi" Caesar dan Scytale ke dalam sebuah aplikasi web untuk mengenkripsi data sebelum disimpan dan mendekripsinya saat ditampilkan.
2. Menganalisis dan mengukur dampak kinerja (overhead) yang disebabkan oleh proses enkripsi dan dekripsi terhadap kecepatan operasi database.
3. Mendemonstrasikan dan memvalidasi bahwa data yang diekstrak melalui simulasi serangan SQL Injection berada dalam format ciphertext yang tidak dapat dibaca.

**Manfaat**

1. Bagi Peneliti: Memperdalam pemahaman mengenai konsep keamanan database, serangan SQL Injection, dan implementasi algoritma kriptografi klasik dalam aplikasi nyata.
2. Bagi Ilmu Pengetahuan: Menyumbangkan data analisis mengenai kinerja dan efektivitas penerapan metode enkripsi klasik sebagai lapisan keamanan tambahan pada database. Hasil penelitian dapat menjadi studi kasus untuk pendekatan keamanan berlapis (*defense-in-depth*).
3. Bagi Masyarakat: Memberikan gambaran dan meningkatkan kesadaran tentang pentingnya enkripsi data di dalam database sebagai langkah mitigasi untuk mengurangi risiko kerugian akibat insiden kebocoran data.

**Metodologi Penelitian**

Penelitian ini akan menggunakan model pengembangan *prototyping* dengan tahapan sebagai berikut:

1. Studi Literatur: Mempelajari secara mendalam tentang serangan SQL Injection, keamanan database, serta konsep dan algoritma kriptografi klasik Caesar Cipher dan Scytale Cipher.
2. Perancangan Sistem: Merancang arsitektur aplikasi web PHP, skema database MySQL, serta alur logika untuk fungsi enkripsi dan dekripsi "Super Enkripsi".
3. Implementasi: Membangun aplikasi web, membuat database, dan mengimplementasikan fungsi-fungsi enkripsi/dekripsi dalam bahasa PHP yang terintegrasi dengan operasi database (INSERT dan SELECT).
4. Pengujian, Pengujian Fungsional: Memastikan proses registrasi, login, penyimpanan, dan penampilan data terenkripsi berjalan dengan benar, Pengujian Kinerja: Mengukur waktu eksekusi query INSERT dan SELECT pada database dengan dan tanpa enkripsi untuk menganalisis *overhead*, Pengujian Keamanan: Melakukan simulasi serangan SQL Injection UNION-based untuk mencoba mengekstrak data dan memverifikasi bahwa data yang terekstrak adalah *ciphertext*.
5. Analisis dan Laporan: Menganalisis data hasil pengujian kinerja dan keamanan, menarik kesimpulan yang menjawab rumusan masalah, dan menyusun laporan skripsi.

**Perbandingan dengan Penelitian Terdahulu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Penulis (Tahun) | Metode | Pembahasan | Hasil |
| 1 | Saputra, A. (2022) | Parameterized Queries | Mencegah SQL Injection dengan memisahkan perintah SQL dari data pengguna di tingkat aplikasi. | Sangat efektif untuk mencegah sebagian besar jenis SQL Injection, namun tidak melindungi data jika database dicuri. |
| 2 | Wibowo, D. (2023) | Enkripsi AES-256 pada Database | Mengimplementasikan enkripsi AES pada kolom-kolom sensitif di database MySQL untuk melindungi data at rest. | Data sangat aman dan terlindungi dari pembacaan tidak sah, namun menyebabkan penurunan kinerja (overhead) yang terukur pada operasi baca/tulis. |
| 3 | Lestari, F. (2021) | Web Application Firewall (WAF) | Menggunakan WAF untuk memfilter lalu lintas HTTP dan memblokir permintaan yang mengandung pola serangan SQL Injection. | Efektif sebagai lapisan pertahanan pertama, namun bisa dilewati (bypassed) oleh serangan yang lebih canggih dan tidak melindungi data di dalam database. |
| 4 | Nugroho, T. (2022) | Analisis Kriptografi Klasik | Menganalisis kekuatan dan kelemahan berbagai sandi klasik (termasuk Caesar dan Vigenere) terhadap serangan modern. | Sandi klasik terbukti sangat rentan terhadap analisis frekuensi dan serangan *brute-force*, tidak cocok untuk data bernilai tinggi. |
| 5 | Hidayat, R. (2024) | Hashing (SHA-256) untuk Kata Sandi | Mengamankan kata sandi pengguna dengan teknik hashing dan salting sebelum disimpan ke database. | Sangat efektif untuk melindungi kata sandi, namun hashing adalah proses satu arah dan tidak dapat digunakan untuk data yang perlu didekripsi kembali. |

Kesimpulan Perbandingan: Penelitian yang ada cenderung fokus pada dua ekstrem: metode pencegahan SQL Injection yang sangat efektif (seperti *Parameterized Queries*) atau penggunaan enkripsi modern yang sangat kuat (seperti AES) yang terkadang berat untuk beberapa aplikasi. Penelitian ini mengambil posisi unik dengan tidak mencoba mencegah serangan, melainkan memitigasi dampaknya menggunakan kombinasi kriptografi klasik yang ringan. Meskipun

Caesar dan Scytale tidak sekuat AES, penelitian ini bertujuan untuk menganalisis kelayakan, kinerja, dan efektivitasnya sebagai lapisan keamanan konseptual yang mudah diimplementasikan, yang menjadi pembeda utama dari penelitian-penelitian sebelumnya.

**Jadwal Penelitian**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kegiatan | **Bulan ke-1** | **Bulan ke-2** | **Bulan ke-3** | **Bulan ke-4** | **Bulan ke-5** | **Bulan ke-6** |
| 1 | Studi Literatur & Finalisasi Proposal | █ |  |  |  |  |  |
| 2 | Perancangan Sistem dan Database | █ | █ |  |  |  |  |
| 3 | Implementasi Aplikasi Web dan Fungsi Enkripsi |  | █ | █ |  |  |  |
| 4 | Pengujian Fungsional dan Kinerja |  |  |  | █ |  |  |
| 5 | Simulasi Serangan dan Pengujian Keamanan |  |  |  | █ |  |  |
| 6 | Analisis Data & Pembahasan Hasil |  |  |  |  | █ |  |
| 7 | Penulisan Kesimpulan & Saran |  |  |  |  | █ |  |
| 8 | Revisi dan Finalisasi Laporan Skripsi |  |  |  |  | █ | █ |
| 9 | Bimbingan dan Konsultasi | █ | █ | █ | █ | █ | █ |
| 10 | Persiapan dan Pelaksanaan Sidang |  |  |  |  |  | █ |

**Referensi**

Abdullah, Z. (2024). *Panduan lengkap pengamanan database: Dari desain hingga pemeliharaan*. Gramedia Pustaka.

Asosiasi Keamanan Informasi Indonesia (AKII). (2024). Laporan tahunan insiden kebocoran data di Indonesia 2023. AKII Press.

Fathoni, A. (2023). Evaluasi penggunaan kriptografi ringan untuk keamanan perangkat Internet of Things (IoT) berdaya rendah. Prosiding Konferensi Nasional Teknik Elektro (KNTE), Surabaya, Indonesia.

Gunawan, I., & Putri, N. (2021). Perbandingan enkripsi tingkat aplikasi dan enkripsi transparan (TDE) pada sistem manajemen database. Jurnal Basis Data dan Informatika (JBDI), 4(1), 22–31.

Hidayat, R. (2024). Mengamankan kata sandi pengguna dengan teknik hashing (SHA-256) dan salting sebelum disimpan ke database. Jurnal Keamanan Informasi dan Siber (JKIS), 5(1), 12–20.

Kurniawan, F. (2024). Pengukuran overhead kinerja pada operasi CRUD akibat enkripsi tingkat aplikasi pada database MySQL. Jurnal Sistem Informasi dan Bisnis Digital (JSISBID), 5(1), 45–55.

Lestari, F. (2021). Implementasi Web Application Firewall (WAF) berbasis signature untuk memblokir serangan SQL Injection. Jurnal Jaringan Komputer dan Keamanan, 3(2), 67–75.

Maulana, R. (2022). Peran Object-Relational Mapping (ORM) dalam mitigasi serangan SQL Injection pada framework Laravel. Jurnal Pengembangan Aplikasi dan Web (JPAW), 3(2), 78–87.

Nugroho, T. (2022). Analisis kekuatan dan kelemahan berbagai sandi klasik (termasuk Caesar dan Vigenere) terhadap serangan modern. Majalah Ilmiah Kriptografi, 2(1), 33–45.

Pratama, I., & Sari, D. (2022). Implementasi visualisasi algoritma kriptografi klasik Caesar dan Scytale untuk media pembelajaran. Jurnal Edukasi dan Teknologi Informasi (JETI), 6(2), 115–124.

Rahman, A., & Santoso, B. (2023). Anatomi serangan SQL Injection dan teknik pertahanan modern. Penerbit Informatika.

Saputra, A. (2022). Pencegahan SQL Injection dengan implementasi parameterized queries pada aplikasi PHP. Jurnal Teknologi Web dan Mobile, 4(1), 56–65.

Subekti, M. (2023). Pemrograman web aman dengan PHP 8: Mencegah kerentanan umum. Media Koding.

Wibowo, D. (2023). Implementasi enkripsi AES-256 pada kolom-kolom sensitif di database MySQL untuk melindungi data at rest. Prosiding Seminar Nasional Ilmu Komputer (SNIK), Jakarta, Indonesia.

Wijaya, E. (2021). Analisis keamanan product cipher berbasis substitusi-transposisi klasik. Prosiding Seminar Nasional Keamanan Siber (SNAKS), Bandung, Indonesia.