BAB 1. PENDAHULUAN

1.1 Latar Belakang Masalah

Pada era digital ini, menjaga keamanan data menjadi sangat penting bagi individu, perusahaan, dan pemerintah. Seiring dengan banyak informasi yang dibagikan dan data yang disimpan secara digital, semakin besar pula risikonya terhadap kerahasiaan dan integritas data. Ancaman seperti peretasan, pencurian data, dan serangan malware semakin parah, sehingga diperlukan solusi keamanan data yang lebih efektif [1]. Dengan meningkatnya nilai informasi sebagai komoditas berharga, penting untuk memastikan bahwa data sensitif hanya dapat diakses oleh individu yang memiliki otorisasi yang tepat. Oleh karena itu, mencegah akses yang tidak sah dan potensi kerusakan bagi pemegang data menjadi prioritas utama [2].

Salah satu bagian terpenting dalam sistem informasi modern adalah basis data. Basis data tidak hanya berfungsi sebagai pusat penyimpanan informasi penting, tetapi juga menjadi target utama dalam berbagai jenis serangan siber karena nilai dan sensitivitas data yang tersimpan didalamnya [3]. *Website* sebagai antarmuka utama dalam banyak sistem digital memberikan kemudahan akses dan interaksi bagi pengguna, namun tidak jarang menjadi titik lemah dari sisi keamanan sistem. Banyak aplikasi web yang tidak dilengkapi dengan mekanisme keamanan yang memadai, sehingga membuka peluang bagi pihak tidak bertanggung jawab untuk mengeksploitasi celah yang ada [4].

Serangan *SQL Injection* merupakan salah satu ancaman serius dalam keamanan aplikasi web. Berbagai metode telah dirancang untuk mendeteksi dan mengantisipasi jenis serangan ini, mulai dari teknik analisis statis hingga pendekatan dinamis. Namun demikian, kedua metode tersebut masih memiliki sejumlah kelemahan. Analisis statis kerap kali tidak mampu mengenali kelemahan dalam proses penyaringan input, khususnya pada sistem yang menerapkan konsep pemrograman berorientasi objek. Di sisi lain, pendekatan dinamis cenderung mereduksi definisi serangan SQLi, sehingga berpotensi memblokir permintaan yang sebenarnya valid. Untuk mengatasi kelemahan ini, salah satu solusi yang

semakin banyak diterapkan adalah enkripsi data. Dengan mengenkripsi data sensitif, proses transmisi informasi menjadi lebih aman dari upaya penyadapan maupun manipulasi, sekaligus memberikan lapisan perlindungan tambahan terhadap akses ilegal [5].

Salah satu pendekatan yang umum digunakan untuk menjaga keamanan data adalah melalui penerapan teknik kriptografi. Kriptografi merupakan cabang ilmu yang mempelajari metode pengamanan informasi agar tetap terlindungi selama proses transmisi dari satu lokasi ke lokasi lainnya. Dengan menerapkan teknik kriptografi yang tepat, data sensitif dapat dienkripsi sehingga hanya pihak-pihak yang memiliki otorisasi yang sah yang dapat mengakses dan memahami informasi tersebut [6].

Melihat pentingnya penerapan kriptografi dalam menjaga kerahasiaan data, muncul kebutuhan untuk mengembangkan pendekatan yang lebih kuat dan berlapis. Pengembangan teknologi enkripsi yang kuat merupakan aspek krusial dalam upaya melindungi informasi penting dari akses pihak-pihak yang tidak bertanggung jawab [1]. Dalam hal ini, algoritma RSA menjadi salah satu metode kriptografi yang banyak digunakan karena menawarkan sistem kunci publik dan privat yang mampu menjaga kerahasiaan dan integritas data [7]. Khususnya, penerapan RSA dengan panjang kunci 2048-bit memberikan tingkat perlindungan tambahan melalui tingkat kompleksitas perhitungan yang tinggi dan prinsip kriptografi asimetris berbasis kunci publik [8].

Penelitian sebelumnya menunjukkan bahwa penerapan algoritma enkripsi yang kuat, seperti RSA, mampu memberikan tingkat perlindungan yang tinggi dalam menjaga kerahasiaan dan integritas data, khususnya pada sistem berbasis web [9]. RSA merupakan algoritma kriptografi yang menggunakan dua pasang kunci, yaitu kunci publik dan kunci privat, dalam proses enkripsi dan dekripsi. Kunci publik dapat disebarluaskan secara terbuka untuk mengenkripsi data, sementara kunci privat hanya diketahui oleh pihak yang berwenang dan digunakan untuk mendekripsi informasi tersebut [10].

Algoritma *Camellia* mendukung panjang kunci yang bervariasi, yaitu 128, 192, dan 256 bit. Modifikasi dari *Feistel Cipher*, algoritma ini menggunakan 18 putaran untuk kunci 128 bit dan 24 putaran untuk kunci 192 dan 256 bit. Meskipun telah

dipatenkan, penggunaan algoritma *Camellia* tidak memerlukan pembayaran royalti asalkan algoritma tersebut tidak dimodifikasi [11].

Penerapan enkripsi dua tahap diyakini mampu memberikan perlindungan data yang lebih kuat dibandingkan pendekatan enkripsi satu tahap, karena mengombinasikan kecepatan enkripsi simetris dan kekuatan autentikasi asimetris dalam satu sistem pengamanan [12]. Pada pendekatan ini, algoritma *Camellia* digunakan terlebih dahulu untuk mengacak isi data guna menjaga kerahasiaan informasi. Selanjutnya, data yang telah dienkripsi tersebut dienkripsi kembali dengan algoritma RSA, yang memanfaatkan pasangan kunci publik dan privat. Pendekatan berlapis ini tidak hanya melindungi data dari akses yang tidak sah, tetapi juga memberikan perlindungan tambahan apabila terjadi serangan terhadap basis data, termasuk serangan *SQL Injection*.

Sebagai implementasi dari pendekatan ini, penelitian ini merancang sebuah program yang mengenkripsi data sensitif seperti Nomor Identitas, Alamat, dan Nama Lengkap menggunakan kombinasi algoritma *Camellia* dan RSA. Sebelum proses enkripsi dilakukan, pengguna diharuskan memasukkan password yang diperlukan untuk proses enkripsi dan dekripsi. Setelah terenkripsi, data kemudian disimpan dan diperbarui dalam basis data.

Penerapan enkripsi pada data pribadi ini sesuai dengan Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang Perlindungan Data Pribadi, yang mewajibkan pengendali data untuk melindungi kerahasiaan dan keamanan data pribadi dengan cara yang sesuai perkembangan teknologi. Nama lengkap, alamat, dan nomor identitas termasuk kategori data pribadi yang wajib dilindungi untuk mencegah penyalahgunaan atau akses ilegal.

1.2 Rumusan Masalah

Seiring meningkatnya ancaman terhadap keamanan data, khususnya pada aplikasi berbasis web, diperlukan sistem yang mampu memberikan perlindungan berlapis terhadap potensi serangan seperti *SQL Injection*. Berdasarkan penjabaran latar belakang yang telah disampaikan, maka perumusan masalah dalam penelitian ini dirumuskan sebagai berikut:

1. Bagaimana mengembangkan mekanisme enkripsi dua tahap sebagai solusi untuk meningkatkan keamanan *Database*.

2. Bagaimana tingkat keberhasilan penggunaan enkripsi dua tahap dengan algoritma *Camellia* dan RSA dalam menjaga keamanan data terhadap potensi serangan.

1.3 Tujuan dan Manfaat

Untuk memberikan arah yang jelas dalam proses perancangan dan implementasi, berikit tujuan yang ingin dicapai serta manfaat yang diharapkan dari penelitian ini :

1. Mengembangkan skema enkripsi dua tahap dengan mengombinasikan algoritma *Camellia* dan RSA untuk meningkatkan ketahanan sistem terhadap potensi serangan *Database*.
2. Menerapkan algoritma *Camellia* dan R cvSA dalam sistem sebagai tahapan enkripsi berurutan yang difungsikan untuk mengamankan *Database*.
3. Menguji efektivitas enkripsi *Camellia* dan RSA dalam menjaga kerahasiaan data.

1.4 Batasan Masalah

Adapun beberapa batasan masalah pada tugas akhir ini adalah sebagai berikut:

1. Penelitian ini hanya berfokus pada implementasi algoritma *Camellia* dan RSA untuk keamanan data pada Sistem Web Data.
2. Pengujian dilakukan dalam kondisi sistem tanpa perlindungan tambahan, dengan menonaktifkan mekanisme keamanan standar seperti SSL/TLS, protokol HTTPS, dan *firewall* website.
3. Pengujian dalam penelitian ini difokuskan secara khusus pada serangan *SQL Injection*.

1.5 Metode Penelitian

1. Studi Literatur – Tahap ini mencakup kajian pustaka terhadap berbagai referensi dan penelitian terdahulu yang berkaitan dengan teknik enkripsi data.
2. Membangun Sistem – Sistem enkripsi dua tahap di bangun dengan mengimplementasikan algoritma *Camellia* sebagai enkripsi tahap pertama dan algoritma RSA sebagai enkripsi tahap kedua pada data yang akan disimpan dalam basis data.

3. Uji Sistem – Sistem yang telah dibangun diuji dengan skenario serangan *SQL Injection.*
4. Analisis Hasil – Hasil pengujian dianalisis berdasarkan ketahanan sistem terhadap serangan dan keberhasilan proses enkripsi-dekripsi.
5. Penyusunan Laporan – Seluruh tahapan penelitian, mulai dari kajian teori, implementasi, hingga pengujian dan analisis, didokumentasikan secara sistematis dalam bentuk laporan tugas akhir.