BAB I
PENDAHULUAN

1.1. Latar Belakang

Pertukaran data merupakan hal yang umum untuk dilakukan di era serba teknologi ini, baik dalam komunikasi pribadi, bisnis, maupun instansi pemerintahan. Setiap saat secara sadar maupun tidak sadar perangkat yang terhubung ke internet akan senantiasa melakukan pertukaran data. Pertukaran data tidak selalu memerlukan campur tangan manusia, beberapa aplikasi yang berjalan pada *background* juga dapat melakukan pertukaran data melalui internet yang seringkali dilakukan guna kepentingan pembaharuan data. *File* juga merupakan sebuah representasi dari data. Salah satu bentuk dari pertukaran data yang sering kali dilakukan adalah pertukaran *file* atau *file sharing*. Bentuk *file* yang dapat dikirimkan juga beragam, mulai dari dokumen atau teks, gambar, video, audio, hingga *file archive* (Guru & Ambhaikar, 2020). Sementara dalam lingkungan akademisi, pengiriman *file* dapat berupa materi kuliah, hasil ujian, hasil penelitian, dan juga presentasi.

Pertukaran *file* dapat dilakukan dalam lingkup individu, kelompok, hingga instansi. Pada umumnya pertukaran *file* dapat dilakukan tanpa menggunakan koneksi internet, contohnya dengan menggunakan *bluetooth.* Namun, pertukaran file menggunakan teknologi tersebut membutuhkan waktu yang lama serta jarak antara pengirim dan penerima yang terbatas (Mahapadi & Das, 2015). Pertukaran *file* menggunakan internet menjadi solusi atas permasalahan tersebut. Dengan menggunakan internet sebagai media pertukaran, maka kecepatan pengiriman dan penerimaan *file* akan bergantung pada kecepatan internet dari pengirim dan penerima yang dipengaruhi oleh jarak, perangkat keras (*hardware*), topologi jaringan beserta ukuran paket data yang dikirimkan (Muttaqin dkk., 2016).

Internet sendiri merupakan sebuah dunia maya jaringan komputer yang menghubungkan miliaran komputer di dunia dengan menggunakan protokol yang terstandarisasi dalam berkomunikasi (Gani, 2018). Dengan banyaknya komputer yang terkoneksi dengan internet, maka memungkinkan untuk menimbulkan masalah baru, salah satunya adalah kebocoran data. Keamanan dan kerahasiaan atas

sebuah data atau informasi merupakan hal yang sangat penting. *File* yang bersifat rahasia harus dijaga kerahasiaannya, sehingga tidak memungkinkan orang lain untuk menerima atau membaca *file* tersebut kecuali pihak yang berwenang. Hal ini tentu menjadi pertimbangan ketika hendak mengirimkan sebuah *file* atau informasi yang bersifat rahasia, dimana integritas sebuah data harus terjaga.

Beberapa contoh kasus kebocoran data yang pernah terjadi di Indonesia adalah kasus kebocoran data yang terjadi pada tahun 2022 pada aplikasi "MyPertamina". Kasus kebocoran data tersebut dilakukan oleh "Bjorka" dan mengantongi data pengguna sebanyak 44.237 juta yang terdiri dari nama, alamat email, nomor induk kependudukan (NIK), nomor pokok wajib pajak (NPWP), nomor telepon, alamat, DOB, gender, pendapatan (per hari, bulan, dan tahun), dan data lainnya (Setjen DPR RI, 2022). Selain itu, kasus kebocoran data juga terjadi pada tahun 2023 yang terjadi pada Bank Syariah Indonesia (BSI) yang diperkirakan sebanyak 15 juta data nasabah tercuri (Setjen DPR RI, 2023). Selain itu, kasus-kasus kebocoran data juga terjadi pada perusahaan-perusahaan besar, seperti pada *Yahoo!* yang terjadi pada tahun 2013. *Hacker* melakukan pencurian data pada perusahaan *Yahoo!* dengan menggunakan metode *email phising*. Email dikirimkan pada seluruh pekerja pada perusahaan tersebut dan ketika terdapat satu pekerja yang terbujuk, maka *hacker* akan memiliki akses di jaringan perusahaan dan dapat memulai aksinya. Selain *Yahoo!* terdapat perusahan-perusahan besar lain yang juga mengalami kasus kebocoran data, yaitu *Facebook*, *Twitter*, dan *LinkedIn*.

Kriptografi merupakan solusi umum yang dapat digunakan untuk mengatasi masalah keamanan data dan salah satunya adalah dalam pertukaran *file*. Kriptografi sendiri merupakan sebuah teknik untuk memberikan kerahasiaan pada sebuah data dengan menggunakan algoritma dan matematika sebagai fokus utamanya (Qadir & Varol, 2019). Dalam kriptografi, data akan diubah kedalam bentuk *ciphertext* sehingga ketika data tercuri maka pencuri tidak dapat mengetahui isi aslinya (Jamaludin, 2018). Meskipun kriptografi mengatasi masalah keamanan data, namun ukuran *file* yang semakin besar dari waktu ke waktu juga menjadi sebuah persoalan tersendiri (Anggraini, 2021). Umumnya algoritma kriptografi asimetris digunakan dalam pertukaran *file*, dikarenakan algoritma kriptografi simetris yang hanya menggunakan satu kunci, sehingga membuatnya kurang aman dalam melakukan

transmisi kunci (Najm dkk., 2020). Kriptografi asimetris memiliki sebuah kelemahan yang terletak pada proses enkripsi dan dekripsinya yang lambat. Hal ini mengakibatkan proses enkripsi dan dekripsi pada *file* semakin lama untuk ukuran *file* yang semakin besar. Maka dari itu, penelitian ini mengusulkan penggunaan *hybrid cryptosystem* dalam mengamankan pengiriman *file*.

Hybrid cryptosystem sendiri merupakan salah satu metode dalam kriptografi yang bekerja dengan menggabungkan algoritma kriptografi simetris (kunci rahasia) dan algoritma kriptografi asimetris (kunci publik) untuk mengamankan data (Rachmawati dkk., 2018). Hasil dari penggabungan kedua jenis algoritma menghasilkan penambahan pada segi keamanan pada proses *key-exchange* dari kriptografi asimetris dan juga pada segi performa (kecepatan) pada proses enkripsi dan dekripsi dari kriptografi simetris (Francis & Monoth, 2018). Beberapa penelitian telah dilakukan untuk menganalisa kelebihan yang diberikan oleh *hybrid cryptosystem*, antara lain adalah penelitian yang dilakukan oleh Tayal dkk. (2017) dengan menggunakan algoritma Huffman Coding dan Hierarchical Encryption Technique yang memberikan peningkatan pada keamanan dan efektifitas pada komputasi, serta penelitian yang dilakukan oleh Mathur dkk. (2016) dengan menggunakan algoritma AES dan ECC yang memberikan keamanan pada pertukaran kunci (*key-exchange*) dan peningkatan keamanan pada *ciphertext* (Francis & Monoth, 2018).

Camellia merupakan salah satu algoritma kriptografi simetris yang tergolong dalam *block cipher*, yang berarti proses enkripsi akan dilakukan dengan membagi data dalam blok-blok dengan panjang yang tetap dan proses enkripsi serta dekripsi dilakukan pada tiap blok tersebut (Čiča, 2016). *Block cipher* merupakan salah satu alasan dalam pemilihan Camellia sebagai algoritma kriptografi simetris pada penelitian ini. Hal ini dikarenakan *file* yang akan dikirimkan dapat dibagi-bagi menjadi blok-blok yang kemudian dilakukan enkripsi maupun dekripsi secara terpisah. Selain itu, pemilihan algoritma Camellia dikarenakan algoritma ini memiliki keamanan yang sepadan dengan *Advanced Encryption Standard* (AES), namun dengan kompleksitas yang rendah (Čiča, 2016). Pemilihan DM-RSA sebagai algoritma kriptografi asimetris pada penelitian ini didasarkan pada keamanan pada algoritma ini yang lebih kuat dibandingkan dengan algoritma RSA

standar. *Dual Modulus* RSA (DM-RSA) sendiri merupakan algoritma kriptografi pengembangan dari algoritma *Rivest Shamir Adleman* (RSA). DM-RSA memberikan peningkatan pada keamanan dengan menggunakan 2 nilai modulus dan 4 kunci yang terdiri atas 2 kunci publik dan 2 kunci pribadi. Hal ini memungkinkan untuk meningkatkan kompleksitas perhitungan pada algoritma RSA dan membuat data lebih tahan terhadap serangan (Manu & Goel, 2017).

Penggunaan metode *end-to-end encryption* (E2EE) memungkinkan untuk meningkatkan keamanan data yang dikirimkan oleh *client*. Hal ini dikarenakan seluruh proses enkripsi dan dekripsi dilakukan pada sisi *client* dan tidak memungkinkan bagi *server* dan juga otoritas lain untuk mengetahui bentuk *file* asli kecuali pemilik kunci rahasia tersebut. Hal ini dikarenakan ketika *file* dalam fase pengiriman atau transmisi, *file* telah terenkripsi atau berbentuk *ciphertext* (Harnal & Chauhan, 2019). Penggunaan enkripsi *end-to-end* sendiri sudah dilakukan oleh banyak perusahaan besar khususnya yang bergerak di bidang penyimpanan *cloud*, seperti BoxCryptor, Icedrive, MEGA, dan Sync (Chen dkk., 2022). Hal ini membuktikan bahwa metode enkripsi *End-to-end* memberikan lapisan keamanan lain untuk meningkatkan keamanan.

Berdasarkan uraian diatas, dapat diambil kesimpulan bahwa pertukaran *file* merupakan hal yang umum dilakukan. Seringkali dalam pengiriman *file*, *file* yang dikirim bersifat rahasia dan memiliki ukuran yang bervariasi. Pemberian keamanan tidak hanya dilakukan untuk *file* yang bersifat rahasia, namun untuk seluruh *file*. Oleh karena itu, penelitian ini mengambil judul "Implementasi *Hybrid Cryptosystem* Menggunakan Camellia Dan *Dual Modulus* RSA Serta Metode E2EE Untuk Pengiriman *File*". Hasil dari penelitian ini diharapkan memudahkan dalam pengiriman *file* yang cepat dan aman. Pemberian dua lapisan keamanan berupa *hybrid cryptosystem* dan metode *end-to-end encryption* (E2EE) memungkinkan integritas data untuk selalu terjaga, khususnya disaat *file* dalam proses pengiriman. Selain itu, hasil dari penelitian ini juga diharapkan dapat memberikan sumbangsih dibidang akademisi dengan memberikan hasil uji coba dari algoritma *hybrid cryptosystem* menggunakan algoritma Camellia dan *Dual Modulus* RSA (DM-RSA).

1.2. Rumusan Masalah

Berdasarkan rincian latar belakang di atas, dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana proses enkripsi dan dekripsi algoritma Camellia dan DM-RSA?
2. Bagaimana pengembangan skema *hybrid cryptosystem* menggunakan algoritma Camellia dan DM-RSA serta metode *end-to-end encryption* (E2EE) untuk pengamanan *file* yang akan diimplementasikan pada aplikasi berbasis *desktop*?
3. Bagaimana perbandingan kecepatan pada skema *hybrid cryptography* menggunakan algoritma Camellia dan DM-RSA?

1.3. Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Melakukan implementasi proses enkripsi dan dekripsi pada algoritma Camellia dan DM-RSA.
2. Merancang dan membangun perangkat lunak untuk pertukaran *file* berbasis *desktop* dengan menggunakan skema *hybrid cryptosystem* menggunakan algoritma Camellia dan DM-RSA serta metode *end-to-end encryption* (E2EE) untuk pengamanan *file*.
3. Melakukan perbandingan kecepatan pada skema *hybrid cryptography* menggunakan algoritma Camellia dan DM-RSA.

1.4. Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Menghasilkan perangkat lunak berbasis *desktop* dengan menerapkan skema *hybrid cryptosystem* menggunakan algoritma Camellia dan DM-RSA serta metode *end-to-end encryption* (E2EE) dalam pengiriman *file*.
2. Memudahkan pengguna dalam melakukan pertukaran *file* secara aman dengan menggunakan skema *hybrid cryptosystem* dan metode *end-to-end encryption* (E2EE).
3. Mengetahui perbandingkan kecepatan pada skema *hybrid cryptography* menggunakan algoritma Camellia dan DM-RSA.

1.5. Batasan Masalah

Adapun batasan masalah yang ditetapkan pada penelitian ini adalah sebagai berikut:

1. Perangkat lunak *client* yang dirancang dan dibangun berupa aplikasi *Graphical User Interface* (GUI) berbasis *desktop*.
2. Proses enkripsi dan pengiriman file untuk *file* lebih dari satu akan dilakukan secara bergantian.
3. Data pengguna disimpan pada *server* dan tidak *persistent*.
4. Tidak membahas sisi keamanan pada proses *client validation* dan *key-exchange*.
5. Tidak melakukan pengujian keamanan.
6. Tidak membahas mengenai metode *key generation*.
7. Kunci untuk kriptografi simetris Camellia sepanjang 128 bit dan menggunakan metode ECB.
8. Kunci untuk kriptografi asimetris DM-RSA sepanjang 128 bit.