BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan jaman, media untuk berkomunikasi ikut mengalami perkembangan. Pada jaman dahulu komunikasi perlu dilakukan secara langsung ataupun melalui media surat untuk bertukar pesan yang mana membutuhkan waktu berhari–hari untuk sampai ke penerima. Dengan perkembangan teknologi, berkomunikasi dapat dilakukan di mana pun dengan memanfaatkan media digital. Waktu yang dibutuhkan untuk bertukar pesan lebih cepat dibandingkan tanpa menggunakan media digital. Media digital yang dapat dipakai untuk komunikasi umumnya adalah *WhatsApp, Line, E-Mail* dan media digital lainnya.

Bentuk pesan yang dapat dikirim melalui media digital mempunyai jenis yang beragam seperti teks, gambar, audio ataupun video. Selain bentuk pesan tersebut, adapula bentuk pesan berupa *coding file*. *Coding file* adalah *file* yang berisi data berupa *script* yang ditulis menggunakan sebuah bahasa pemrograman untuk digunakan sebagai instruksi kepada komputer untuk melakukan suatu tugas (Hardikushwawa, 2020). Saling kirim *coding file* ini biasanya terjadi pada bidang IT khususnya *programmer*. *Programmer* mengirim *coding file* yang telah dibuat dengan tujuan untuk diperbaharui oleh *programmer* lainnya.

Pengiriman *coding file* menggunakan server publik beresiko karena terdapat kemungkinan *file* yang dikirim dapat diretas. Oleh karena itu, keamanan pesan merupakan aspek yang penting dalam pengiriman ini terutama jika *coding file* yang dikirim bersifat rahasia. Sebelum pesan dikirim melalui jaringan publik, dibutuhkan pengamanan *file* agar jikalau *file* tersebut diambil orang lain, *file* tersebut tidak dapat dipakai oleh orang yang tidak diberi kewenangan. Salah satu cara untuk menjaga kerahasiaan *file* ini yaitu dengan mengubah isi *coding file* mejadi pesan yang tersamarkan menggunakan algoritma kriptografi.

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (Schneier, 2010). Pada dasarnya kriptografi terdiri dari proses enkripsi dan dekripsi. Proses untuk menyamarkan pesan asli disebut dengan enkripsi. Sebaliknya, proses untuk mengembalikan pesan tersamarkan menjadi pesan asli disebut dengan dekripsi.

Kedua proses tersebut dilakukan dengan menggunakan suatu kunci dari algoritma kriptografi. Terdapat dua jenis algoritma kriptografi berdasarkan kuncinya. Algoritma kriptografi simetris menggunakan kunci yang sama untuk proses enkripsi dan dekripsinya, sedangkan algoritma kriptografi asimetris menggunakan kunci yang berbeda pada proses enkripsi dan dekripsi.

Salah satu algoritma yang bisa digunakan untuk menyamarkan *coding file* adalah algoritma *One Time Pad* (OTP). Algortima OTP merupakan algoritma simetris yang sangat aman untuk digunakan menilik kunci yang tidak berulang dan panjangnya sama dengan jumlah panjang pesan (Firdaus dkk. 2017). Proses enkripsi dan dekripsi algoritma OTP menggunakan konsep operasi XOR. Setiap karakter pesan dan kunci diubah menjadi kode angka dalam bentuk bit lalu berdasarkan urutan karakter pesan di-XOR-kan dengan kuncinya masing-masing. Namun, algoritma ini mempunyai kekurangan yaitu, dibutuhkan waktu yang lama untuk menentukan kunci yang akan dibentuk sepanjang pesan (Firdaus dkk. 2017). Pada bagian besar data pun memiliki kekurangan yaitu semakin besar data yang akan disamarkan, maka semakin besar juga data kunci algoritma OTP yang dibutuhkan. Oleh karena itu, dibutuhkan modifikasi agar algoritma OTP tidak memerlukan data yang sama besar dengan pesan namun tetap memiliki kunci sepanjang pesan.

Modifikasi yang dapat dilakukan untuk mengatasi pemasalahan besar data pada kunci OTP yaitu dengan cara membangkitkan bilangan acak secara terus-menerus sedemikian sehingga jumlah bilangan acak yang terbentuk sama dengan jumlah panjang pesan. Bilangan acak yang terbentuk merupakan kunci OTP yang dapat di-XOR secara berurutan dengan pesan yang akan diamakan.

Terdapat beberapa penelitian terdahulu mengenai pembangkitan bilangan acak. Firdaus dkk. (2017) melakukan penelitian pada pembangkitan kunci OTP menggunakan barisan fibonacci menghasilkan barisan kunci yang tidak berulang. Pada penelitian tersebut ditunjukkan juga bahwa algoritma OTP dapat digabungkan dengan algoritma lainnya yaitu *Affine Cipher*. Biantara dkk. (2015) melakukan penelitian mengenai modifikasi *Linear Congurential Genetar* (LCG) untuk optimalisasi hasil acak yang menghasilkan metode *Couple Linear Congurential Genetar* (CLCG). Triwibowo & Ariyus (2020) melakukan penelitian mengenai

penerapan CLCG pada algoritma OTP menghasilkan kunci OTP berupa bilangan acak yang tidak berulang.

Berdasarkan penelitian terdahulu, metode *Couple Linear Congurential Genetar* (CLCG) diambil untuk diimplementasikan pada algoritma OTP. Metode ini dipilih karena metode CLCG terbukti menghasilkan bilangan acak dan membutuhkan variabel pembangkitan yang lebih banyak daripada metode lainnya yaitu fibonacci dan *Linear Congurential Genetar* (LCG). Pemilihan variabel yang lebih banyak ini dapat memperkecil kemungkinan pihak tidak berwenang untuk dapat menebak semua variabel pembangkitan dengan benar.

Algoritma OTP modifikasi CLCG menggunakan kunci yang sama untuk proses enkripsi dan dekripsi sehingga dibutuhkan jalur pertukaran kunci OTP CLCG yang sangat aman untuk menjaga keamanan *coding file*. Jika kunci diketahui oleh orang lain maka orang tersebut dapat mendekripsi *file* tersandi kembali ke *file* aslinya. Oleh karena itu, algoritma OTP CLCG ini akan digabungkan dengan algoritma asimetris. Algoritma asimetris dipilih karena kunci pada proses dekripsi hanya dimiliki oleh pihak penerima sehingga jikalau kunci OTP CLCG diketahui pihak lain, *file* tersandi tetap aman dan tidak dapat didekripsi tanpa kunci dekripsi pada algorimta asimetris. Salah satu algortima asimetri yang dapat digunakan yaitu *Rivest Shamir Adleman* (RSA).

RSA adalah algoritma asimetris yang dianggap memiliki keamanan cukup bagus dilihat dari sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima (Simargolang, 2017). Namun, walaupun tingkat keamanannya cukup tinggi proses enkripsi dan dekripsinya cukup lama dikarenakan oleh algoritma RSA itu sendiri. Semakin besar kunci RSA yang digunakan maka, semakin lama juga proses enkripsi dan dekripsinya. Oleh karena itu, algoritma ini perlu dimodifikasi. Salah satu algortima yang dapat mempercepat proses dekripsi pada RSA adalah *Chinese Remainder Theorem* (CRT). CRT adalah algoritma yang berfungsi untuk mengurangi perhitungan aritmatika modular dengan modulus besar sehingga proses dekripsi akan lebih cepat (Arief & Saputra, 2016). Dengan menggunakan CRT, permasalahan lama waktu dekripsi pada algoritma OTP CLCG dan RSA dapat dipersingkat.

Terdapat beberapa penelitian terdahulu mengenai RSA beserta penggabungan RSA dengan algoritma lainnya. Arief & Saputra (2016), melakukan penelitian yang pada *instan messaging* yang mana hasilnya menunjukkan bahwa algoritma RSA-CRT lebih cepat dalam proses dekripsinya jika dibandingkan dengan algoritma RSA. Ristiana (2017) melakukan penelitan mengenai penggabungan algoritma OTP dan RSA di mana penggabungan kedua algoritma dilakukan secara *hybrid*. Sitohang dkk. (2019) melakukan penelitian mengenai penggabungan algoritma *Simplified Elliptic Curve Integrated Encryption Scheme* (S-ECIES) dan RSA yang mana hasil kriptosistemnya dapat mempersulit peretasan.

Berdasarkan penjabaran tersebut, penulis akan melakukan penelitian mengenai pengamanan *coding file* menggunakan algoritma OTP modifikasi CLCG dan RSA *Chinese Remainder Theorem*. Pada penelitan sebelumnya algoritma kriptografi simetris OTP dan asimetris RSA bersifat *hybrid* dimana kriptografi asimetris berperan dalam pengamanan kunci algoritma kriptografi simetris. Pada penelitian ini, kedua algortima yang dimodifikasi dipakai untuk mengenkripsi *coding file* yang akan diamankan. Diharapkan penelitian ini dapat menambah pengetahuan mengenai enkripsi dan dekripsi *coding file* menggunakan algoritma OTP CLCG dan RSA-CRT.

1.2 Batasan Masalah

Pada penelitian ini diberikan batasan masalah yaitu proses enkripsi dan dekripsi hanya dilakukan pada *coding file* yang dapat dibuka menggunakan *text editor*, antara lain *python*, *c++* dan matlab.

1.3 Rumusan Masalah

Berdasarkan latar belakang di atas, terdapat beberapa rumusan masalah yang akan dikaji pada penelitian ini, yaitu:

1. Bagaimana teknik melakukan enkripsi dan dekripsi pesan menggunakan Algoritma OTP CLCG dan RSA-CRT?
2. Bagaimana konstruksi aplikasi keamanan *coding file* menggunakan algoritma OTP CLCG dan RSA-CRT menggunakan bahasa pemrograman *Python*?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini adalah sebagai berikut:

1. Mendeskripsikan proses enkripsi dan dekripsi pesan menggunakan algoritma OTP CLCG dan RSA-CRT
2. Mengkonstruksi aplikasi keamanan *coding file* menggunakan algoritma OTP CLCG dan RSA-CRT menggunakan bahasa pemrograman *Python*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Manfaat Teoritis

   Secara teoritis penelitian ini memiliki manfaat dalam memberikan pengetahuan mengenai cara enkripsi dan dekripsi *coding file* menggunakan algoritma OTP CLCG dan RSA-CRT.

2. Manfaat Praktis

   Secara praktis penelitian ini memiliki manfaat berupa aplikasi pengamanan *coding file* menggunakan algoritma OTP CLCG dan RSA-CRT yang diharapkan dapat digunakan di kemudian hari.