BAB III
METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan studi literatur, pengembangan model kriptosistem gabungan dan implementasi model ke dalam program komputer, dengan rincian:

3.1 Identifikasi Masalah

Seiring dengan perkembangan teknologi informasi dan komunikasi, kriptografi memainkan peranan penting dalam mengatasi masalah-masalah terkait keamanan data. Semakin berkembangnya teknologi informasi dan komunikasi, maka semakin cepat pula perkembangan kriptanalisis. Kriptanalis sekarang ini dapat memecahkan algoritma yang sudah ada dengan cepat. Salah satu cara untuk mempersulit kriptanalisis terhadap sebuah *cipher* adalah dengan menggabungkan dua kriptosistem. Dalam penelitian ini akan dilakukan penggabungan kriptosistem S-ECIES dengan kriptosistem RSA.

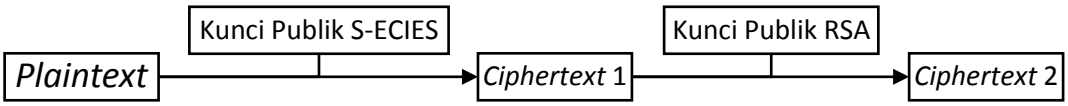
3.2 Model Dasar

Model dasar yang digunakan adalah kriptosistem S-ECIES dan kriptosistem RSA. Kriptosistem S-ECIES merupakan penyederhanaan dari kriptosistem *Elliptic Curve Integrated Encryption Scheme* (ECIES) yang menggunakan ECC untuk membentuk kunci yang akan digunakan dalam kriptografi simetris. Dalam S-ECIES, *plaintext* disembunyikan ke dalam titik-titik pada kurva eliptik (*masking*). Keamanan kriptosistem ini didasarkan pada sulitnya memecahkan *Elliptic Curve Discrete Logarithm Problem* (ECDLP). Kriptosistem RSA ditemukan oleh Ron Rivest, Adi Shamir dan Leonard Adleman pada tahun 1977. Keamanan kriptosistem ini didasarkan pada sulitnya memfaktorkan suatu bilangan menjadi dua buah bilangan prima.

3.3 Pengembangan Model

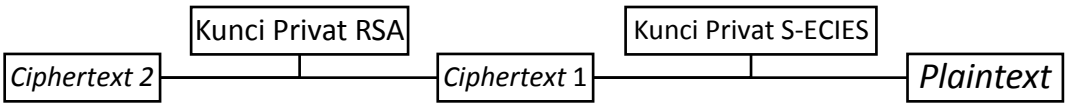
Pada penelitian ini akan dirancang sebuah kriptosistem gabungan S-ECIES dan RSA. Tujuannya adalah untuk membentuk sebuah kriptosistem baru yang

tetap aman. Langkah pertama adalah mengenkripsi *plaintext* menggunakan kunci publik S-ECIES. Kemudian, hasil *ciphertext* akan dienkripsi lagi menggunakan kunci publik RSA. Skema enkripsi dapat dilihat pada gambar berikut.



Gambar 1.1 Skema Enkripsi Model Pengembangan

Untuk proses dekripsi, *ciphertext* didekripsi menggunakan kunci privat RSA. Kemudian, hasil dekripsi akan didekripsi lagi menggunakan kunci privat S-ECIES. Skema dekripsi dapat dilihat pada gambar berikut:



Gambar 1.2 Skema Dekripsi Model Pengembangan

Untuk mempermudah proses enkripsi dan dekripsi, dilakukan pembuatan program komputer.

3.4 Perancangan Program Komputer

Program yang akan dibuat memiliki tampilan awal (menu utama) yang memiliki tiga pilihan, yaitu pembangkitan kunci, enkripsi dan dekripsi. Proses pembangkitan kunci menggunakan kurva eliptik berukuran 256-bit yang telah disetujui oleh *National Institute of Standards and Technology* (kurva P-256) dan algoritma RSA yang menggunakan modulus berukuran 2048-bit. Hasil pembangkitan kunci kemudian disimpan menjadi dua buah file *object* Python yang masing-masing berisi kunci publik dan kunci privat. Input dari proses enkripsi berupa *plaintext* yang akan dienkripsi dan file *object* Python kunci publik dengan output berupa *ciphertext* disimpan ke dalam sebuah file *object* Python. Sementara dalam proses dekripsi, input berupa file *object* Python yang berisi

ciphertext dan file *object* Python kunci privat dengan output berupa *plaintext* hasil dekripsi.

3.5 Pembuatan Program

Program kriptosistem gabungan S-ECIES dan RSA akan dibuat menggunakan bahasa pemrograman Python versi 3.5. Penggunaan Python dikarenakan bahasa pemrograman Python dapat menampung bilangan bulat yang sangat besar. Selain itu, Python mendukung *Object Oriented Programming* dan dapat menyimpan *object* sebagai *binary file* sehingga *ciphertext* dapat dibuat menjadi sebuah *object* dan kemudian dapat disimpan sebagai *file*.

3.6 Validasi

Pada tahap ini dilakukan validasi terhadap *output* program yang diperoleh. Validasi berupa pengecekan hasil *plaintext* yang sudah mengalami proses enkripsi dan dekripsi apakah sama dengan *plaintext* aslinya atau tidak.

3.7 Penarikan Kesimpulan

Hasil dari penggabungan dua kriptosistem ini adalah sebuah kriptosistem baru yang lebih sulit untuk dipecahkan, karena kriptanalis harus memecahkan dua masalah dengan tingkat kesulitan sangat tinggi untuk berhasil mendapatkan *plaintext* dari *ciphertext*.