BAB V
KESIMPULAN DAN SARAN

Berdasarkan rumusan masalah dan pembahasan hasil penelitian yang telah dipaparkan pada bab sebelumnya maka diperoleh kesimpulan dan saran dari hasil penelitian tersebut.

5.1. Kesimpulan

Dari hasil penelitian yang dipaparkan pada bab sebelumnya, diperoleh kesimpulan sebagai berikut.

1. Skema program pengamanan *big data* dengan AES-256 dan kunci *hashing* SHA-256 dimulai dengan adanya perubahan kunci yang dipakai. Pada algoritma AES-256 yang standar, kunci yang diinput hanya diubah dalam bentuk heksadesimal berdasarkan tabel ASCII yang kemudian akan digunakan pada proses enkripsi atau pun dekripsi AES-256. Namun, pada penyandian algoritma AES-256 dengan kunci *hashing* SHA-256 ini, kunci yang digunakan untuk enkripsi adalah kunci yang telah di *hashing* menggunakan SHA-256. *Message digest* yang dihasilkan akan memiliki jumlah lebih dari yang diperlukan sehingga *message digest* yang dihasilkan akan dipangkas sesuai kebutuhan.
2. Program pengamanan *big data* dengan AES-256 dan kunci *hashing* SHA-256 dilakukan mengonstruksi program dengan Bahasa pemrograman *python* dibantu dengan modul *tkinter* untuk membuat tampilan program. Pada program aplikasi tersebut, pengguna dapat mengenkripsi dan mendeskripsi *file*. Hasil enkripsi data akan menghasilkan *file ciphertext* berupa karakter acak yang memiliki ukuran lebih besar dari *file* aslinya. Ukuran *file plaintext* yang cukup besar menjadi alasan dibutuhkannya alokasi memori yang cukup besar untuk menjalankan program dengan baik tanpa adanya kendala.

Tatyana Citra Diantari, 2024
PENGAMANAN BIG DATA DENGAN ADVANCED ENCRYPTION STANDARD-256 DAN KUNCI HASHING SECURE HASH ALGORITHM-256
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

5.2. Saran

Ada beberapa saran yang dapat penulis berikan untuk penelitian selanjutnya, yaitu

1. Mengalokasikan memori yang besar sebelum membuat program enkripsi dan dekripsi *big data*.
2. Melakukan penelitian untuk *file* yang lebih besar tanpa menggunakan *library* kriptografi.
3. Melakukan pengembangan untuk pengamanan *big data* pada sebuah server secara *real-time*.
4. Melakukan kompresi data sehingga *file* yang digunakan tidak akan memiliki ukuran yang begitu besar sehingga dapat meminimalisir kegagalan karena kekurangan memori yang dialokasikan pada program.