BAB I
PENDAHULUAN

1.1 Latar Belakang

Dengan munculnya internet, perangkat-perangkat menjadi saling terhubung lebih luas lagi. Istilah yang menggambarkan keterhubungan antar perangkat melalui internet ini disebut dengan *Internet of Thing* (IoT) (Çamur, 2020). Keamanan menjadi masalah utama dalam pengembangan IoT (Thakor, dkk., 2021) sehingga dibutuhkan suatu ilmu, yaitu Kriptografi, yang dapat menjawab permasalahan tersebut (Bhardwaj, dkk., 2017). Kriptografi dapat menjadi solusi untuk masalah keamanan karena Schneier dalam Munir (2019) menyebutkan bahwa kriptografi bertujuan untuk memberi layanan keamanan di antaranya kerahasiaan (*confidentiality*), integritas data (*data integrity*), autentikasi (*authentication*) dan anti-penyangkalan (*non-repudiation*). Di antara keempat layanan keamanan tersebut, fokus utama pada penelitan ini adalah kerahasiaan (*confidentiality*).

Untuk mengamankan IoT, algoritma kriptografi diimplementasikan ke dalam perangkat IoT. Namun yang menjadi permasalahan adalah beberapa algoritma, contohnya AES, tidak dapat diimplementasikan ke beberapa perangkat IoT meskipun AES aman terhadap serangan kriptanalisis yang sudah diketahui (Çamur, 2020). Hal ini dikarenakan beberapa IoT memiliki keterbatasan baterai, RAM, ROM, kebutuhan *latency* yang kecil maupun kemampuan komputasi yang tak sebaik komputer desktop (Çamur, 2020; Moriai & Katagi, 2008; Thakor, dkk., 2021). Oleh karena itu, sebuah keluarga kriptografi yang baru yaitu kriptografi *lightweight* menjadi solusi untuk permasalahan tersebut (Çamur, 2020; Priyanka dalam Sallam dan Behesti, 2018).

Beberapa contoh algoritma yang termasuk ke dalam kriptografi *lightweight* antara lain Trivium, *Present*, Ascon, dan Gift. Algortima kriptografi *lightweight* yang termasuk ke dalam kategori *block ciphers* antara lain *Present*, Ascon, dan Gift sedangkan Trivium termasuk ke dalam kategori *stream ciphers* (Çamur, 2020; Putri, dkk., 2022).

Setiap algoritma memiliki keunggulannya masing–masing. Biryukov dan Perrin (2017) berpendapat bahwa pengukuran performa suatu algoritma dapat dibedakan menjadi dua kategori, yaitu *Hardware Case* dan *Software Case.*

Hardware Case terbagi lagi menjadi *memory consumption, throughput, latency,* dan power *consumption. Software Case* terbagi lagi menjadi *RAM consumption*, *code size* dan *throughput.*

Menurut Thakor, dkk., (2021) *Present Cipher* sangat efisien dari segi *hardware* maupun *software*. Hal ini sejalan dengan Diehl, dkk. (2017) yang mengungkapkan bahwa Speck, Twine dan *Present Cipher* efisien dari segi *hardware* maupun *software.* Berbeda dengan Speck dan Twine, *Present Cipher* telah distandarisasi oleh ISO/IEC (29192-2:2019). Lebih lanjut menurut Çamur (2020) *Present Cipher* merupakan contoh terbaik dari algoritma kriptografi *lightweight* sehingga dari pernyataan-pernyataan tersebut, *Present Cipher* menjadi algoritma kriptografi *lightweight* yang sangat menarik untuk dikaji.

Present Cipher termasuk ke dalam blok cipher dengan skema SPN (Bogdanov, dkk., 2007). Jika *Present Cipher* diterapkan untuk pengamanan pesan teks maka dibutuhkan mode operasi karena *Present Cipher* hanya memiliki input sebesar 64-bit sedangkan pesan teks dapat melebihi 64-bit. Pada penelitian ini, mode operasi yang digunakan adalah CBC (*Cipher Block Chaining*). Menurut Bujari & Aribas (2017) CBC telah umum digunakan dan tidak seperti mode operasi ECB (*Electronic Code Books*), CBC tahan terhadap serangan substitusi.

Sudah ada banyak penelitian terdahulu yang mengkaji *Present Cipher*, diantaranya Diehl, dkk. (2017) yang membandingkan efisiensi *Present Cipher* dengan algoritma lain dari segi *hardware* dan *software,* Bellizia, dkk. (2016) yang meneliti implementasi dan analisis kerentanan *Present*-80 *Cipher* terhadap serangan *side channel*, dan Lara-Nino (2017) yang meneliti implementasi *hardware* dari *Present Cipher*.

Berbeda dari penelitian-penelitian terdahulu, pada penelitian ini *Present Cipher* digunakan untuk mengamankan suatu pesan teks. Pesan teks diamankan dengan cara menyamarkan pesan sehingga tidak dapat dipahami oleh pembaca. Pesan teks diamankan menggunakan aplikasi yang dibuat menggunakan bahasa pemrograman Python dengan *Present Cipher* sebagai algoritma yang mengamankan pesan teks tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dibuat dua rumusan masalah sebagai berikut:

1. Bagaimana pengamanan pesan teks dengan Kriptografi *Lightweight Present Cipher* pada *Cipher Block Chaining*?
2. Bagaimana implementasi pengamanan pesan teks dengan Kriptografi *Lightweight Present Cipher* pada *Cipher Block Chaining* dalam aplikasi dengan bahasa program Python?
3. Bagaimana memvalidasi program aplikasi pengamanan pesan teks dengan Kriptografi *Lightweight Present Cipher* pada *Cipher Block Chaining*?

1.3 Tujuan Penelitian

Tujuan yang hendak dicapai dalam penelitian ini adalah:

1. Memperoleh pengamanan pesan teks dengan Kriptografi *Lightweight Present Cipher* pada *Cipher Block Chaining*.
2. Mengimplementasikan pengamanan pesan teks dengan Kriptografi *Lightweight Present Cipher* pada *Cipher Block Chaining* dalam aplikasi untuk menggunakan bahasa pemrograman Python.
3. Memvalidasi program aplikasi untuk pengamanan pesan teks dengan Kriptografi *Lightweight Present Cipher* pada *Cipher Block Chaining.*

1.4 Manfaat Penelitian

Manfaat yang hendak dicapai dari penelitian ini adalah:

1. Penelitian ini diharapkan berkontribusi dalam pengembangan metode terkait pengamanan pesan teks terutama pada kriptografi *lightweight Present Cipher* pada *Cipher Block Chaining*.
2. Memperoleh aplikasi pengamanan pesan teks dengan kriptografi *lightweight Present Cipher* pada *Cipher Block Chaining* menggunakan bahasa Python sehingga dapat digunakan oleh *user*.

1.5 Batasan Masalah

1. Pesan merupakan karakter-karakter alfabet, angka, simbol dan tanda baca (*printable character*).
2. Cipherteks merupakan karakter-karakter heksadesimal.

3. Pesan input untuk aplikasi diketik langsung pada aplikasi atau pesan disimpan pada file dengan format ".txt"