BAB II
LANDASAN TEORI

1.1. Bitlocker Drive Encryption

BitLocker Drive Encryption adalah sebuah fitur enkripsi satu cakram penuh yang terdapat di dalam sistem operasi Microsoft yang didesain untuk melindungi data dengan melakukan enkripsi terhadap keseluruhan partisi. *BitLocker Drive Encryption* menggunakan algoritma AES dalam mode *Code Block Chaining* (CBC) dengan panjang kunci 128-bit, yang digabungkan dengan Elephant diffuser untuk meningkatkan keamanannya.

Bitlocker Drive Encryption memiliki tiga modus operasi, yaitu *Transparent Operation Mode, User Authentication Mode*, dan *USB Key Mode*. Ketiga jenis mode operasi ini menentukan bagaimana *Bitlocker Drive Encryption* dioperasikan dan tingkat keamanan yang ditawarkan setiap mode implementasi berbeda-beda. Bitlocker menggunakan media penyimpanan eksternal sebagai media penyimpanan kunci. Media tersebut dapat berupa media storage USB maupun sebuah chip bernama *Trusted Platform Module (TPM).*

1.1.1. Modus operasi transparan

Modus ini menggunakan sepenuhnya kemampuan perangkat keras *TPM* 1.2 untuk memberikan keamanan yang tinggi dengan kenyamanan kepada pengguna-pengguna dapat masuk *log* ke Windows secara normal, seolah tidak ada proses enkripsi berlangsung di dalamnya. Kunci yang digunakan untuk melakukan enkripsi akan dienkripsi dengan menggunakan *chip TPM* dan hanya akan dibuka kepada kode pemuat sistem operasi jika berkas-berkas yang dibutuhkan dalam rangka proses *booting* terlihat belum dimodifikasi. Komponen BitLocker sebelum OS berjalan ini mampu melakukannya dengan mengimplementasikan metodologi *Static Root of Trust Measurement,* yang didefinisikan oleh *Trusted Computing Group*.

1.1.2. Modus autentikasi pengguna

Modus ini mengharuskan pengguna memasukkan beberapa parameter autentikasi dalam lingkungan sebelum melakukan proses *booting* agar mampu melakukan proses *booting* ke sistem operasi. Modus ini memiliki dua metodologi, yakni dengan menggunakan *Personal Identification Number (PIN)* yang dimasukkan oleh pengguna, atau perangkat *USB* yang dimasukkan oleh pengguna yang mengandung kunci yang dibutuhkan.

1.1.3. Modus *USB Key*

Pengguna harus memasukkan perangkat *USB* yang mengandung kunci yang dibutuhkan agar computer mampu melakukan *booting* terhadap sistem operasi yang dilindungi oleh BitLocker. Modus ini memang tidak membutuhkan *TPM*, tapi modus ini membutuhkan BIOS yang mampu membaca perangkat *USB* sebelum *booting*.

1.2. TPM (Trusted Platform Module)

Trusted Platform Module (TPM) adalah sebuah *microchip* yang terintegrasi dengan *motherboard* komputer dan sifatnya unik pada setiap *motherboard* yang berbeda. *TPM* memiliki fungsionalitas khusus untuk menangani aspek keamanan komputer melalui kriptografi, pada *Bitlocker Drive Encryption*, *TPM* hanya dilibatkan sebatas penyimpan kunci saja.

1.3. Algoritma Kriptografi

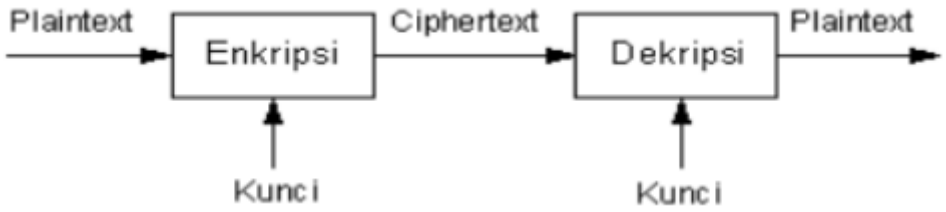
1.3.1. Pengertian algoritma kriptografi

Algoritma merupakan urutan langkah - langkah logis untuk menyelesaikan masalah yang disusun secara matematis dan benar. Kriptografi (*cryptography*) berasal dari kata “crypto” yang berarti *secret* (rahasia) dan “graphy” yang berarti *writing* (tulisan). Kriptografi merupakan suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Algoritma kriptografi

merupakan langkah - langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut.

1.3.2. Fungsi dasar algoritma kriptografi

a. Enkripsi merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut *plaintext* yang diubah menjadi kode - kode yang tidak dimengerti. Enskripsi bisa diartikan dengan *cipher* atau kode.
b. Dekripsi merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (teskasli), disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan algoritma untuk enkripsi.
c. Kunci, yang dimaksud adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*). Secara umum fungsi tersebut digambarkan :



Gambar 2. 1 Proses Enkripsi dan Dekripsi ((Purba, 2019)

1.3.3. Jenis Algoritma Kriptografi

Berdasarkan kunci yang dipakainya, Algoritma Kriptografi dibagi menjadi tiga jenis :

a. Algoritma Simetri

Algoritma yang memakai kunci simetri diantaranya adalah :

1) Blok Chiper : Data Encryption Standard (DES), Internasional Data Encryption Algorithm (IDEA), Advanced Encryption Standard (AES).
2) Stream Chiper : On Time Pad (OTP), A5, RC2, RC4, RC5, dan RC6.

b. Algoritma Asimetri

Algoritma yang memakai kunci *public* di antaranya adalah : *Digital Signature Algorithm (DSA), RSA, DiffleHellman (DH), Elliptic Curve Cryptography (ECC),* Kriptografi Quantum, dan lain sebagainya.

c. Fungsi Hash

Contoh algoritma yang menggunakan fungsi hash adalah MD5 dan SHA1.

1.4. WinImage

WinImage adalah *suite* pencitraan *disk* yang lengkap untuk memudahkan pembuatan, membaca, dan mengedit banyak format gambar dan sistem file, termasuk *DMF, VHD, FAT, ISO, NTFS,* dan *Linux*. Gambar disk adalah salinan persis dari disk fisik (*floppy, CD-ROM, hard disk, USB, disk VHD,* dan lain - lain.) atau partisi yang mempertahankan struktur aslinya. WinImage dapat membuat ulang gambar *disk* pada *hard drive* atau media lain, melihat kontennya, mengekstrak file berbasis gambar, menambahkan file dan direktori baru, mengubah format, dan mendefrag gambar.

Program ini memiliki banyak kegunaan utilitarian di rumah dan di kantor. Pengguna *PC* yang serius, mungkin memiliki banyak *floppy disk* lama tapi masih berguna. WinImage dapat mengubahnya menjadi gambar disk, yang dapat disimpan di hard drive dan dibuat ulang. Dikombinasikan dengan alat pembuat *CD*, WinImage dapat membantu dalam membuat *disk boot* kustom sendiri dengan diagnostik perangkat keras atau perangkat lunak pembersih *virus* untuk mengembalikan dan menjalankan *PC* bermasalah tanpa berada di Windows. WinImage memungkinkan dapat menghemat waktu berjam-jam dan bahkan berhari-hari untuk memulihkan sistem dan konfigurasi pada mesin yang mengalami kerusakan hard disk atau kerusakan perangkat lunak. Kemampuan ini adalah suatu keharusan untuk kelas pelatihan, di mana memulihkan konfigurasi PC yang rusak dengan cepat sangat penting (Vollant, 2005).

Fitur yang ada dalam WinImage :

a. Buat image disk dari drive yang dapat dilepas (seperti *drive USB), CD-ROM, floppy*

b. Ekstrak file dari gambar *disk*
c. Buat gambar *disk* kosong
d. Menyuntikkan file dan direktori ke dalam gambar *disk* yang ada
e. Mengubah format gambar *disk*
f. Defrag gambar *disk*
g. Mode "*Batch assistant*" yang kuat yang memungkinkan Anda mengotomatiskan banyak operasi,

1.5. Passware Kit Forensic

Passware Kit Forensic adalah solusi penemuan bukti elektronik terenkripsi lengkap yang melaporkan dan mendekripsi semua item yang dilindungi kata sandi di komputer. Perangkat lunak ini mengenali 300+ jenis file dan bekerja dalam mode batch memulihkan kata sandi (Passware, 2021).

1.6. Brute Force

1.6.1. Pengertian Algoritma Brute Force

Brute force adalah sebuah pendekatan yang langsung (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*).

1.6.2. Kelebihan Algoritma *Brute Force*

Berikut ini beberapa kelebihan yang dimiliki oleh *brute force*, yaitu:

a. Algoritma *brute force* dapat digunakan untuk memecahkan hampir sebagian besar masalah.
b. Sederhana dan mudah dimengerti.
c. Menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, perkalian matriks.
d. Menghasilkan algoritma baku (standar) untuk tugas-tugas komputasi seperti penjumlahan/perkalian N buah bilangan, menentukan elemen minimum atau maksimum ditabel.

1.6.3. Kelemahan Algoritma *Brute Force*

Berikut ini beberapa kelemahan yang dimiliki oleh *brute force*, yaitu:

a. Jarang menghasilkan algoritma yang mangkus/efektif.
b. Lambat sehingga tidak dapat diterima.
c. Tidak sekreatif teknik pemecahan masalah lainnya.

1.6.4. Cara Kerja Algoritma *Brute Force*

Berikut ini beberapa langkah cara kerja yang dimiliki oleh *brute force*, yaitu:

a. Mula-mula string dicocokkan pada awal teks.
b. Dengan bergerak dari kiri kekanan, dibandingkan setiap karakter di dalam string dengan karakter yang bersesuaian di dalam teks, jika sesuai dibandingkan tersebut mengeluarkan hasil.
c. Jika string belum ditemukan kecocokan dari teks belum habis, maka geser string satu karakter ke kanan dan berulang langkah ke 2.

Penelitian tentang *Brute Force Attack* sudah banyak diteliti oleh orang yang mahir di bidang keamanan jaringan, salah satunya tentang "*Brute Force Attack* dan Penerapanya Pada *Password Cracking*" yang ditulis oleh Krisnaldi Eka Pramudita, penelitian yang dilakukan mengulas tentang algoritma *brute force* dalam lingkup teknologi informasi dan penerapannya dalam membobol atau meretas sebuah *password* misalnya *password* untuk *login* facebook atau wordpress. Algoritma *brute force* yang umumnya dipakai untuk meretas kasus *password* seperti ini umumnya disebut *Brute Force Attack.*

Brute force attack menggunakan sebuah himpunan karakter atau teks yang akan dipakai untuk referensi karakter-karakter dari *password* yang ingin dibobol/diretas. Himpunan karakter yang dipakai akan menjadi sebuah ukuran keefektifan dari algoritma itu sendiri. Semakin banyak anggota himpunan karakter ini, tentunya persentasi *password cracking* untuk sebuah *password* dapat diretas akan meninggi. Namun, makin banyak karakter yang ada di dalam himpunan itu harus dibayar dengan waktu pengerjaan yang lebih lama. *Brute Force* ini sudah

mulai dikembangkan untuk meretas *password*. Pengembangannya adalah *dictionary attack* yang menggunakan algoritma *brute force* tetapi himpunan karakternya berasal dari sebuah kamus (misalnya KBBI) sehingga memungkinkan untuk memangkas waktu yang diperlukan *Brute Force Attack* pada umumnya walaupun ber-*drawback* tidak ditemukannya *password* (Pramudita, 2011).

Penelitian terdahulu tentang "implementasi alogaritma br*ute force* dalam pencarian data katalog buku perpustakaan" yang membahas tentang pola pencocokan kumpulan karakter huruf/kata (selanjutnya disebut String) yang satu dengan String yang lainnya menggunakan Algoritma *Brute Force* atau biasa disebut Algoritma Naïf (Naïve Algorithm) dan cara penyelesaian masalah pencocokan pola String dengan Algoritma *Brute Force* tergolong termasuk cara penyelesaian yang tidak cerdas karena memiliki cara kerja yang sederhana, *String Matching* merupakan salah satu algoritma yang digunakan untuk mempercepat proses pencarian kata yang diinginkan. *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic* atau *statistical matching*. Algoritma *string matching* telah sering digunakan sebelumnya seperti contoh pada proses pencocokan *string* berdasarkan persamaan teks data yaitu *Brute Force*. Algoritma *brute force* dipilih karena algoritma ini dapat digunakan untuk melakukan pencarian *string* atau teks (Mesran, 2014).

Berdasarkan pada penelitian selanjutnya, dari jurnal yang ditulis oleh Oni Rafizan yang terkait dengan "Analisa penyerangan social enginering", maka dilakukan pengembangan penelitian yang membahas tentang analisa sebuah serangan *brute force* dengan program aplikasi dari sistem operasi kali linux yaitu Aplikasi *Scanning* yang nantinya akan melakukan serangan terhadap sebuah *webserver* yang sudah di rancang sebelumnya dan penelitian ini menghasilkan sebuah analisa yang nantinya dapat membantu para pengguna *website* yang terkena serangan brute force untuk mengetahui bagaimana cara kerja proses *brute force* berlangsung. WPScan dan Sql Ijection adalah *scanner* keamanan yang memeriksa keamanan *Website* menggunakan metode "*blackbox*" fiturnya yaitu untuk pencacahan *username* dan *multithreaded password* untuk proses

bruteforcing, pencacah versi *plugin Website* dan pencacahan kerentanan sistem (Aprina, 2013).

Serangan *brute-force* adalah sebuah teknik serangan terhadap sebuah sistem keamanan komputer yang menggunakan percobaan terhadap semua kunci yang mungkin. Pendekatan ini pada awalnya merujuk pada sebuah program komputer yang mengandalkan kekuatan pemrosesan komputer dibandingkan kecerdasan manusia. Contoh, untuk menyelesaikan sebuah persamaan kuadrat seperti $x^2+7x-44=0$, di mana x adalah sebuah *integer*, dengan menggunakan teknik serangan *brute force*, penggunanya hanya dituntut untuk membuat program yang mencoba semua nilai integer yang mungkin untuk persamaan tersebut hingga nilai x sebagai jawabannya muncul. Istilah brute force sendiri dipopulerkan oleh Kenneth Thompson, dengan mottonya: "*When in doubt, use brute-force*" (jika ragu, gunakan *brute-force*), secara sederhana menebak *password* dengan mencoba semua kombinasi karakter yang mungkin. *Brute force attack* digunakan untuk menjebol akses ke suatu *host (server/workstation/network)* atau kepada data yang terenkripsi. Metode ini dipakai para cracker untuk mendapatkan *account* secara tidak sah, dan sangat berguna untuk memecahkan enkripsi. Enkripsi macam apapun, seperti *Blowfish, AES, DES, Triple DES*, dsb secara teoritis dapat dipecahkan dengan *brute-force attack*. Pemakaian *password* sembarangan, memakai *password* yang Cuma sepanjang 3 karakter, menggunakan kata kunci yang mudah ditebak, menggunakan *password* yang sama, menggunakan nama, memakai nomor telepon, sudah pasti sangat tidak aman. Namun *brute force attack* bisa saja memakan waktu bahkan sampai berbulan-bulan atau tahun bergantung dari bagaimana rumit passwordnya (Rafizan, 2012).

1.7. Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu referensi dalam melakukan sebuah penelitian, tujuannya agar dapat memperkaya teori yang digunakan dalam tinjauan penelitian. Berdasarkan penelusuran sebelumnya, tidak ditemukan penelitian dengan nama yang sama dengan penelitian ini. Adapun pengutipan dari

beberapa penelitian yang digunakan sebagai referensi untuk memperkaya literatur penelitian.

Berikut adalah gambaran dari penelitian sebelumnya berupa beberapa ulasan terkait dengan penelitian yang dilakukan yaitu:

Tabel 2. 1 Penelitian Terdahulu

Nama Peneliti	**Judul Penelitian**	**Hasil Penelitian**
Fresly Nandar Pabokory, Indah Fitri Astuti, Awang Harsa Kridalaksana	Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma *Advanced Encryption Standard*	Tujuan penelitiannya untuk membuat sebuah sistem keamanan data dengan mengimplementasikan kriptografi pada pesan teks, isi file dokumen, dan file dokumen dengan melakukan perhitungan algoritma *Advanced Encryption Standard (AES).*
Kesimpulan: Penelitian tersebut mencapai tujuan yang telah ditetapkan untuk menciptakan sebuah sistem keamanan data dengan mengimplementasikan kriptografi pada pesan teks, isi file dokumen, dan file dokumen dengan melakukan perhitungan algoritma *Advanced Encryption Standard (AES)* dengan menambahkan pendukung kemanan steganografi dalam penyembunyian pesan teks atau file dalam file citra		
Sumber: (Pabokory, Astuti, & Kridalaksana, 2015)		
Nama Peneliti	**Judul Penelitian**	**Hasil Penelitian**
Mohammad Hamdani dan Dwi Darmi Sa'diyahti	Aplikasi Credant dan BitLocker untuk Sistem Keamanan Data	Tujuan dari penelitian ini ialah implementasi dan pengujian sistem keamanan data menggunakan aplikasi

	Computer	Credant dan BitLocker untuk melindungi data/informasi pada suatu komputer dari serangan atau usaha pengambilan data oleh pihak-pihak yang tidak diinginkan.
Kesimpulan: Implementasi Credant dan BitLocker pada suatu komputer, data/informasi yang tersimpan menjadi lebih aman dari serangan/usaha akses data yang tidak diinginkan secara langsung maupun melalui jaringan baik di *level user* maupun *hardware*, karena data telah terenkripsi menggunakan Credant pada *level user* dan *hard disk* telah terenkripsi seluruhnya menggunakan BitLocker.		
Sumber: (Hamdani & Sa'diyahti, 2015)		
Nama Peneliti	**Judul Penelitian**	**Hasil Penelitian**
G. Harianto, E. B. Setiawan dan Y. R. Murti	*Automated social media account identification using Simplified Brute Force*	Penelitian ini melakukan *crawl* data ke beberapa media sosial sekaligus menggunakan DOM Parser. Data yang terkumpul dilakukan pencocokan menggunakan algoritma *Brute Force* yang disederhanakan dengan efisiensi waktu 78,16% di mana Facebook mendapatkan nilai Akurasi tertinggi yaitu 90% pada identifikasi akun yang belum diverifikasi dan Instagram mendapatkan nilai akurasi 80% untuk identifikasi akun terverifikasi.
Kesimpulan: *Crawling* data dari dua sosial media yaitu Facebook dan Instagram mendapatkan nilai pencocokan dengan akurasi yang tinggi dengan menggunakan Algoritma *Brute Force*. Hasil yang didapatkan ialah 90% nilai akurasi Facebook dan 80% nilai akurasi Instagram.		
Sumber: (Setiawan, Harianto, & Murti, 2019)		
Nama Peneliti	**Judul Penelitian**	**Hasil Penelitian**

Sugiharto	Implementasi Algoritma *Brute Force* Dalam Pencarian Kebudayaan Di Indonesia Berbasis *Mobile Application*	Penelitian ini memanfaatkan teknologi perangkat *mobile smartphone* berbasis android untuk membangun aplikasi pencarian informasi kebudayaan yang terdapat di Indonesia menggunakan algoritma *brute force.*
Kesimpulan: Penerapan algoitma *brute force* pada aplikasi pencarian kebudayaan di Indonesia berhasil diterapkan sehingga dapat menyelesaikan masalah dalam melakukan pencarian data budaya, karena algoritma ini menemukan data yang dicari.		
Sumber: (Sugiharto, 2018)		
Nama Peneliti	**Judul Penelitian**	**Hasil Penelitian**
Bayu Widia Santoso, Firdiansyah Sundawa, Muhammad Azhari	Implementasi Algoritma *Brute Force* Sebagai Mesin Pencari (*Search Engine*) Berbasis *Web* Pada *Database*	Penelitian ini bertujuan untuk membentuk sebuah metode pencarian atau *search engine* menggunakan algoritma *brute force*
Kesimpulan: Algortima *brute force* berhasil diterapkan sebagai *search engine* dalam sebuah aplikasi yang membantu dokter dan perawat untuk megetahui jenis obat dan zat yang terkandung dalam obat tersebut.		
Sumber: (Santoso, Sundawa, & Azhari, 2016)		