BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. Enkripsi

Enkripsi merupakan salah satu metode yang ada dalam kriptografi yaitu teknik untuk mengubah data asli (*plain text*) menjadi data yang hanya bisa dibaca oleh pemilik kunci saja (*encrypted text*) (Wicaksana and Setiawan, 2020).

2.1.2. Dekripsi

Dekripsi merupakan kebalikan dari enkripsi, yaitu teknik untuk mengubah kembali data yang sudah terenkripsi (*encrypted text*) menjadi data semula (*plain text*) (Wicaksana and Setiawan, 2020).

2.1.3. OpenSSL

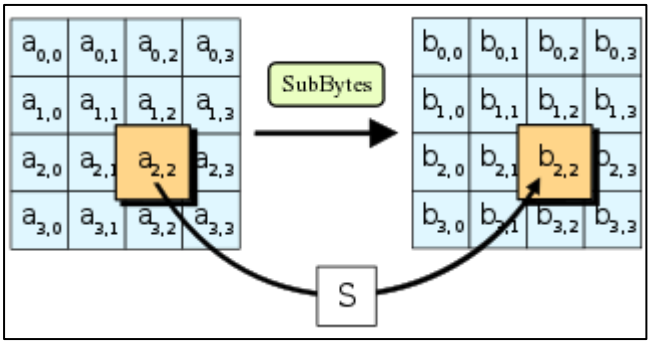
OpenSSL adalah sebuah perpustakaan sumber terbuka *SSL/TLS (Secure Socket Layer / Transport Layer Security)* yang populer, efektif serta paling banyak digunakan sebagai protokol untuk mengamankan komunikasi antar jaringan guna menyediakan integritas serta kerahasiaan data antara dua aplikasi yang berbeda (R, Suresh and Pradeep, 2013).

2.1.4. Algoritma AES

AES (*Advanced Encryption Standard*) merupakan algoritma kriptografi simetris dengan panjang blok *cipher* 128 bit dan memiliki 3 varian kunci yaitu 128 bit, 192 bit, dan 256 bit (Hameed, Ibrahim and Manap, 2018; Pasaribu et al., 2018; Renuka Devi, Suba Rani and Noble Mary Juliet, 2019; Riyaldhi, Rojali and

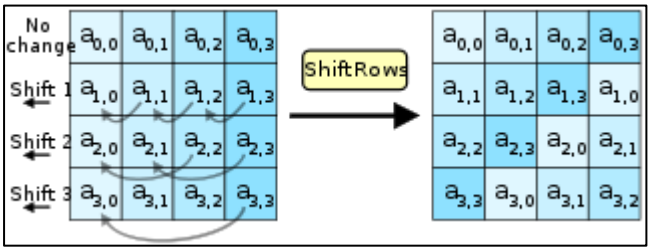
Kurniawan, 2017; Tulloh, Permanasari and Harahap, 2016; Wiharto and Irawan, 2018).

Algoritma AES didesain dengan menggunakan 4 transformasi yaitu *Subbyte Transformation*, *Shift Row Transform*, *Mix Column Transform*, dan *Add Round Key* (Hameed, Ibrahim and Manap, 2018; Nuari and Ratama, 2020; Rawal, 2016; Utami, Arifudin and Alamsyah, 2019) serta merupakan keluarga dari blok *cipher*.



Gambar 2.1 Transformasi *SubBytes*

Gambar 2.1 menunjukan bahwa pada tiap bit ai,j "status" ditukar dengan SubBytes S (ai,j) dengan kotak substitusi 8 bit. Langkah ini memberi sifat *nonlinear* dalam penyandian ini.



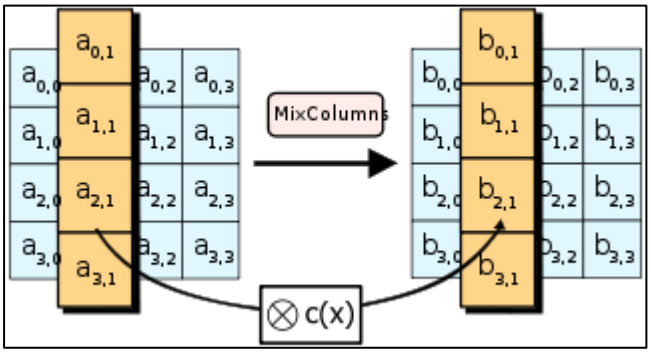
Gambar 2.2 Transformasi *ShiftRows*

Gambar 2.2 menunjukan transformasi ShiftRows yang mengubah tiap baris dengan uraian sebagai berikut.

1. Bagian AES, baris pertama dibiarkan.

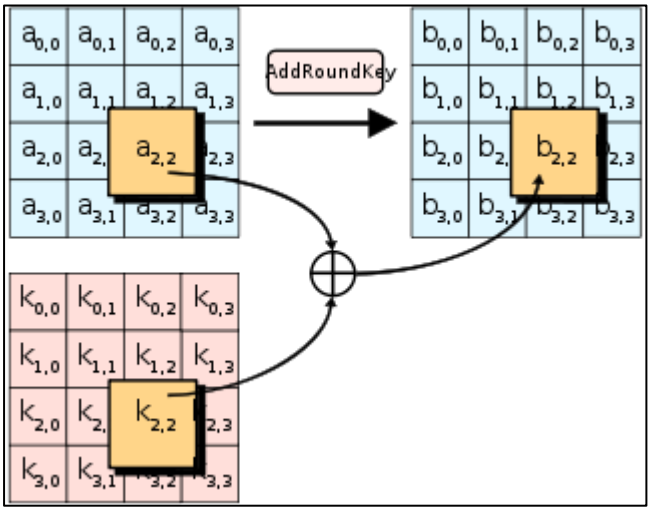
2. Tiap bit pada baris kedua digeser sekali.
3. Pada baris ketiga dan keempat pun digeser sekali.
4. Baris ketiga digeser dua kali dan baris keempat digeser tiga kali.

Langkah-langkah yang telah diuraikan di atas dapat dijalankan agar tiap kolom tidak dienkripsi masing-masing.



Gambar 2.3 Transformasi *MixColumns*

Gambar 2.3 menunjukan bahwa empat bit dalam tiap kolom "status" digabung dengan transformasi *linear* yang ditukar. Fungsi MixColumns menerima empat bit masukan dan mengeluarkan empat bit yang tiap bit masukannya saling memengaruhi. Fungsi ini bersama dengan ShiftRows memberikan penghamburan dalam penyandian.



Gambar 2.4 Transformasi *AddRoundKey*

Gambar 2.4 menunjukan bahwa sub kunci digabung dengan "status". Tiap ronde sebuah sub kunci dibuat dari kunci utama dengan penjadwalan kunci AES. Tiap sub kunci berukuran sama dengan "status". Sub kunci ditambahkan menggunakan penggabungan tiap bit "status" dan bit yang letaknya sama pada sub kunci dengan operasi XOR.

2.1.5. Blok Cipher

Blok *cipher* digunakan pada algoritma kriptografi simetris dan membutuhkan kunci untuk melakukan proses enkripsi maupun dekripsi. Blok *cipher* mempunyai 4 mode operasi yaitu ECB (*Electronic-Code-Book*), CBC (*Cipher-Block-Chaining*), CFC (*Cipher-Feedback*), dan OFB (*Output-Feedback*) (Murdowo, 2019). Tahun 2001 NIST (*National Institute of Standards and Technology*) mempublikasikan algoritma AES dan menambah 1 mode baru yaitu CTR (*Counter*) (Dworkin, 2005).

Mode *Electronic-Code-Book* (ECB) adalah mode kerahasiaan yang menampilkan kunci yang diberikan, penugasan blok *ciphertext* tetap ke setiap blok

plain text, analog dengan penugasan kata kode dalam buku kode. Mode Electronic Codebook (ECB) didefinisikan pada gambar 2.5:

$$C_j = CIPH_K(P_j)$$

Gambar 2.5 Enkripsi mode ECB

Gambar 2.5 menunjukan bahwa fungsi *forward cipher* diterapkan secara langsung dan independen ke setiap blok *plain text*. Urutan blok keluaran yang dihasilkan adalah *ciphertext*.

$$P_j = CIPH^{-1}_K(C_j)$$

Gambar 2.6 Dekripsi mode ECB

Gambar 2.6 menunjukan bahwa fungsi *reverse cipher* diterapkan secara langsung dan independen ke setiap blok *ciphertext*. Urutan blok keluaran yang dihasilkan adalah *plain text*.

Mode *Cipher Block Chaining* (CBC) adalah mode kerahasiaan yang proses enkripsinya menampilkan penggabungan (“chaining”) blok *plain text* dengan blok *ciphertext* sebelumnya. Mode CBC membutuhkan IV untuk digabungkan dengan blok *plain text* pertama.

$$\begin{aligned} C_1 &= CIPH_K(P_1 \oplus IV); \\ C_j &= CIPH_K(P_j \oplus C_{j-1}) \end{aligned}$$

Gambar 2.7 Enkripsi mode CBC

Gambar 2.7 menunjukan blok masukan pertama dibentuk dengan XOR blok pertama *plain text* dengan IV. Fungsi *forward cipher* diterapkan pada blok masukan pertama, dan blok keluaran yang dihasilkan adalah blok pertama dari *ciphertext*.

$$P_1 = CIPH^{-1}_K(C_1) \oplus IV;$$
$$P_j = CIPH^{-1}_K(C_j) \oplus C_{j-1}$$

Gambar 2.8 Dekripsi mode CBC

Gambar 2.8 menunjukan rumus untuk memulihkan setiap blok *plain text* (kecuali yang pertama), fungsi *reverse cipher* diterapkan ke blok *ciphertext* yang sesuai, dan blok yang dihasilkan dikecualikan dengan blok *ciphertext* sebelumnya.

Mode *Cipher Feedback* (CFB) adalah mode kerahasiaan yang menampilkan umpan balik dari segmen *ciphertext* berturut-turut ke dalam blok masukan *forward cipher* untuk menghasilkan blok keluaran yang di XOR dengan *plain text* untuk menghasilkan *ciphertext,* dan sebaliknya.

$$I_1 = IV;$$
$$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$$
$$O_j = CIPH_K(I_j)$$
$$C^{\#}_j = P^{\#}_j \oplus MSB_s(O_j)$$

Gambar 2.9 Enkripsi mode CFB

Gambar 2.9 menunjukan bahwa blok masukan pertama adalah IV, dan operasi *forward cipher* diterapkan ke IV untuk menghasilkan blok keluaran pertama. Segmen *ciphertext* pertama dihasilkan dengan meng XOR segmen *plain text* pertama dengan bit paling signifikan dari blok keluaran pertama.

$$I_1 = IV;$$
$$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$$
$$O_j = CIPH_K(I_j)$$
$$P^{\#}_j = C^{\#}_j \oplus MSB_s(O_j)$$

Gambar 2.10 Dekripsi mode CFB

Gambar 2.10 menunjukan bahwa IV adalah blok masukan pertama, dan setiap blok masukan yang berurutan dibentuk seperti pada enkripsi CFB, dengan menggabungkan bit paling signifikan dari blok masukan sebelumnya dengan bit paling signifikan dari *ciphertext* sebelumnya.

Mode *Output Feedback* (OFB) adalah mode kerahasiaan yang menampilkan iterasi *forward cipher* pada IV untuk menghasilkan urutan blok keluaran yang di XOR dengan *plain text* untuk menghasilkan *ciphertext*, dan sebaliknya.

$$
\begin{aligned}
&I_1 = IV;\\
&I_j = O_{j-1}\\
&O_j = CIPH_K(I_j)\\
&C_j = P_j \oplus O_j\\
&C^*_n = P^*_n \oplus MSB_u(O_n).
\end{aligned}
$$

Gambar 2.11 Enkripsi mode OFB

Gambar 2.11 menunjukan bahwa IV ditransformasikan oleh fungsi *forward cipher* untuk menghasilkan blok keluaran pertama. Blok keluaran pertama di XOR dengan blok *plain text* pertama yang menghasilkan blok *ciphertext* pertama.

$$
\begin{aligned}
&I_1 = IV;\\
&I_j = O_{j-1}\\
&O_j = CIPH_K(I_j)\\
&P_j = C_j \oplus O_j\\
&P^*_n = C^*_n \oplus MSB_u(O_n).
\end{aligned}
$$

Gambar 2.12 Dekripsi mode OFB

Gambar 2.12 menunjukan bahwa IV ditransformasikan oleh fungsi *forward cipher* untuk menghasilkan blok keluaran pertama. Blok keluaran pertama di XOR dengan blok *ciphertext* pertama yang memulihkan blok *plain text* pertama.

Mode *Counter* (CTR) adalah mode kerahasiaan yang menampilkan aplikasi *forward cipher* ke satu set blok masukan, yang disebut *counter*, untuk menghasilkan urutan blok keluaran yang di XOR dengan *plain text* untuk menghasilkan *ciphertext*, dan sebaliknya. sebaliknya.

$$O_j = CIPH_K(T_j)$$
$$C_j = P_j \oplus O_j$$
$$C_n^* = P_n^* \oplus MSB_u(O_n).$$

Gambar 2.13 Enkripsi mode CTR

Gambar 2.13 menunjukan bahwa fungsi *forward cipher* dipanggil pada setiap blok counter, dan blok keluaran yang dihasilkan di XOR dengan blok *plain text* yang sesuai untuk menghasilkan blok *ciphertext*.

$$O_j = CIPH_K(T_j)$$
$$P_j = C_j \oplus O_j$$
$$P_n^* = C_n^* \oplus MSB_u(O_n).$$

Gambar 2.14 Dekripsi mode CTR

Gambar 2.14 menunjukan bahwa fungsi *forward cipher* dipanggil pada setiap blok *counter*, dan blok keluaran yang dihasilkan di XOR dengan blok *ciphertext* yang sesuai untuk memulihkan blok *plain text*.

2.1.6. Transmisi Data

HTTP (*Hypertext Transfer Protocol*) adalah protokol lapisan aplikasi untuk mengirimkan dokumen hypermedia, seperti *HTML (Hypertext Markup Language)*. *HTTP* dirancang untuk komunikasi antara *client* dan *server*, tetapi juga dapat digunakan untuk tujuan lain (MDN Contributors, 2019). Transmisi data merupakan proses perpindahan data dari suatu sumber ke sumber lainnya.

2.1.7. Man In The Middle

Serangan *man-in-the-middle* (MITM) adalah serangan yang diam-diam mentransfer dan mungkin mengubah korespondensi antara dua pihak yang percaya mereka langsung berkomunikasi satu sama lain. Serangan *Man-in-the-middle* (MITM) adalah istilah umum ketika pelakunya memposisikan dirinya dalam diskusi antara klien dan aplikasi untuk mendengarkan diam-diam atau untuk meniru satu pihak, membuatnya tampak seolah-olah pertukaran informasi biasa sedang berlangsung (Mallik et al., 2019).

2.1.8. Entropi *Shannon*

Dalam teori informasi, entropi *Shannon* adalah teknik untuk mengukur jumlah ketidakteraturan, yaitu informasi yang terkandung dalam segmen tertentu. Dia menghasilkan jumlah "informasi" dan "keacakan" dalam data yang diberikan dalam jumlah bit per sampel (Paul, 2017).

$$H(X) = -\sum_{i=1}^{n} p(X_i) \log_2 p(X_i)$$

Gambar 2.15 Rumus entropy shannon

Gambar 2.15 menunjukan rumus untuk mencari entropy *shannon* dimana X merupakan himpunan, n merupakan jumlah partisi X, $p(X_i)$ merupakan proporsi X_i terhadap X.

2.2. Penelitian Terkait dan Kebaruan Penelitian

2.2.1. State of The Art Advanced Encryption Standard

Penelitian terkait yang digunakan sebagai acuan pada penelitian ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 State of The Art Advanced Encryption Standard

State of The Art – Advanced Encryption Standard (AES)		
Judul	Kontribusi dan Hasil Utama	Batasan
Combination of Advanced Encryption Standard 256 Bits with MD5 to Secure Documents on Android Smartphone (Pasaribu et al., 2018).	● Implementasi enkripsi dan dekripsi berkas pada perangkat android. ● Menggunakan md5 sebagai hash untuk membuat kunci. ● Menggunakan AES-256	● Implementasi digunakan di perangkat android.
Enkripsi Data Menggunakan *Advanced Encryption Standard* 256 (Wiharto and Irawan, 2018).	● Aplikasi mudah digunakan karena mempunyai tampilan yang sederhana. ● Memiliki tingkat kesulitan enkripsi 32 bit. ● Menggunakan AES-256	● Belum bisa mengenkripsi dan dekripsi berkas dengan format .avi, .mkv,.mp3 dan lain sebagainya secara maksimal. ● Belum bisa mengenkripsi dengan ukuran data yang lebih besar dengan kunci enkripsi yang lebih banyak.
Kriptografi *Advanced Encryption Standard* (AES) untuk Penyandian File Dokumen (Tulloh, Permanasari and Harahap, 2016).	● Enkripsi dan dekripsi di bantu oleh MATLAB sehingga mempercepat proses dan menghasilkan keluaran yang tepat .	● Jumlah karakter tidak bisa melebihi 16 karakter. ● Hanya bisa melakukan enkripsi dan dekripsi pada teks.

Tabel 2.2 Lanjutan Tabel 2.1

An Image Encryption & Decryption and Comparison with Text - AES Algorithm (Renuka Devi, Suba Rani and Noble Mary Juliet, 2019).	● Menggunakan kunci 128bit. ● Cocok digunakan untuk aplikasi *real-time.* ● Mengirim informasi melalui gambar lebih dapat diandalkan daripada menggunakan teks.	● waktu enkripsi dan dekripsi meningkat dengan bertambahnya ukuran file.
Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA (Hermawan and Ujianto, 2021).	● Menyisipkan pesan yang sudah dienkripsi menggunakan AES dan RSA dan di sisipkan ke media gambar dengan teknik steganografi LSB	● Hanya bisa menyisipkan pesan rahasia pada media gambar
Implementasi Algoritma Kriptografi AES (*Advanced Encryption Standard*) 128 Bit untuk Pengamanan Dokumen Shipping (Nuari and Ratama, 2020).	● Dapat diakses melalui web	● Masih menggunakan kunci 128 Bit
Using AES Algorithm Encryption and Decryption of Text File, Image and Audio in Openssl and Time Calculation for Execution (Anand, 2020).	● Menggunakan mode CBC. ● Menggunakan *CLI* untuk melakukan proses enkripsi dan dekripsi.	● Hanya bisa mengenkripsi teks, gambar, dan audio.
Implementasi Algoritma AES-256 untuk Pengamanan Layanan API pada Restful dengan Autentikasi Json Web Tokens (Ardiansyah et al., 2019).	● Implementasi pada REST. ● Menggunakan AES-256.	● Studi kasus di Perusahaan Bintang Cemerlang Abadi.

Tabel 2.3 Lanjutan Tabel 2.1

A Comparison of the 3DES and AES Encryption Standards (Aleisa, 2015).	● AES lebih unggul daripada 3DES.	● Membandingkan 3DES dengan AES.
AES Based Text Encryption Using 12 Rounds with Dynamic Key Selection (Mathur and Bansode, 2016).	● Menggabungkan AES dengan *Elliptic Curve Encryption (ECC).*	● Menggunakan AES-192
Extended AES Algorithm with Custom Encryption for Government-level Classified Messages (Dasgupta and Das, 2019).	● Menggabungkan AES dengan *Caesar Cipher.* ● Menggunakan huruf terakhir sebagai kunci untuk mengenkripsi dengan AES.	● Menggunakan huruf awal untuk mengenkripsi dengan AES.
Implementasi Algoritma Advanced Encryption Standard (AES) 128 untuk Enkripsi dan Dekripsi File Dokumen (Prameshwari and Sastra, 2018).	● Implementasi pada aplikasi desktop.	● Menggunakan AES-128. ● Jenis file .pdf, .doc, .txt
Double Encrypted Key Based AES Combined Coding For Improved Cloud Security (Srilakshmi and Bhargavi, 2020).	● Membandingkan kompresi Huffman dan LZMA. ● Kompresi LZMA lebih baik daripada Huffman.	● Menggunakan AES-256
A Comparative Analysis of AES Common Modes of Operation (Almuhammadi and Al-Hejri, 2017).	● Membandingkan blok *cipher* yang digunakan pada AES	● Mencari blok *cipher* dengan performa terbaik untuk AES
Kombinasi Algoritma Kriptogarafi AES dan DES untuk Enkripsi File Dokumen Proposal (Irawan et al., 2020).	● Dapat diakses melalui web. ● Menggunakan AES-128.	● Jenis file .docx, .doc, .pdf, .xls, .xlsx, .ppt, .pptx.

2.2.2. Keterbaruan Penelitian

Keterbaruan dalam penelitian ini adalah mengintegrasikan algoritma AES-256 mode *CTR* dan OpenSSL untuk proses enkripsi maupun dekripsi untuk meningkatkan keamanan data pada proses transmisi.