ANALISIS IMPLEMENTASI ENKRIPSI DAN DEKRIPSI MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) PADA JAVA NETBEANS

Dina Eka Fitriani[1], Natasha Zulatifa[2], Dyah Putri Anggraini[3], Indrawan Ady Saputro[4]
[1234]STMIK Amikom Surakarta
[1234]Sukoharjo Indonesia
Email: [1] dina.10388@mhs.amikomsolo.ac.id,
[2]natasha.10398@mhs.amikomsolo.ac.id, [3] dyah.10385@mhs.amikomsolo.ac.id,
[4]indrawanadysaputro@gmail.com

Abstract

Information security is very important in the modern computing world. Securing data, including images, is very important because of the sensitivity and value of the data contained therein. Cryptography is a field that studies how data is protected. There are two categories of cryptography: classic and modern. AES will become the main standard for encrypting all types of electronic data used in commercial applications such as telecommunications, personal, banking and financial information, as well as financial transactions. In this research practice, the Advanced Encryption Standard (AES) algorithm is used to encrypt and decrypt messages using Java. AES uses the same key for both encryption and decryption processes, ensuring that only the person who has the key can access the actual information. There are many important steps that must be taken before implementing this, one of which is creating a secret key using the AES algorithm which produces encryption: PBKDF2WithHmacSHA256 and using a Cipher object for the encryption and decryption process. This method is successfully used in research practice to encrypt the original "password" text into ciphertext and then decrypt it back to the original text.

Keywords: *Kriptografi, AES, Java, Algoritma Advanced Encryption, Cipher, Enkripsi, Dekripsi, Password, CipherText, Komersial, Telekomunikasi.*

Abstraksi

Keamanan informasi sangat penting dalam dunia komputasi modern. Pengamanan data, termasuk gambar, sangat penting karena sensitivitas dan nilai data yang terkandung di dalamnya. Kriptografi adalah bidang yang mempelajari bagaimana data dilindungi. Ada dua kategori kriptografi: klasik dan modern. AES akan menjadi standar utama untuk mengenkripsi semua jenis data elektronik yang digunakan dalam aplikasi komersial seperti telekomunikasi, informasi pribadi, perbankan, dan finansial, serta transaksi finansial. Dalam praktik penelitian ini, algoritma Advanced Encryption Standard (AES) digunakan untuk mengenkripsi dan mendekripsi pesan menggunakan Java. AES menggunakan kunci yang sama untuk kedua proses enkripsi dan dekripsi, memastikan bahwa hanya orang yang memiliki kunci tersebut yang dapat mengakses informasi yang sebenarnya. Banyak langkah penting yang harus dilakukan sebelum

melakukan implementasi ini, salah satunya adalah membuat kunci rahasia menggunakan algoritma AES yang menghasilkan enkripsi: PBKDF2WithHmacSHA256 dan menggunakan objek Cipher untuk proses enkripsi dan dekripsi. Metode ini berhasil digunakan dalam praktik penelitian untuk mengenkripsi teks asli "password" ke dalam teks ciphertext dan kemudian mendekripsinya kembali ke teks aslinya.

Kata Kunci: *Kriptografi, AES, Java, Algoritma Advanced Encryption, Cipher, Enkripsi, Dekripsi, Password, CipherText, Komersial, Telekomunikasi.*

1. PENDAHULUAN

Dalam dunia komputasi modern, keamanan informasi merupakan hal yang sangat penting. Pengamanan data, termasuk gambar, menjadi perhatian utama karena sensitivitas dan nilai informasi yang disimpan di dalamnya. Kriptografi adalah bidang yang menyelidiki metode yang digunakan untuk melindungi data. Kriptografi ada dua jenis: kriptografi klasik dan modern. Kriptografi klasik menggunakan satu kunci untuk menyembunyikan data dan menggunakan dua teknik dasar: subsitusi dan transposisi. Kriptografi kontemporer memerlukan algoritma yang kompleks, tetapi ini karena algoritma modern menggunakan komputer. Untuk menyelesaikan masalah tersebut, proses pengamanan data harus dianalisis dan dikembangkan, dengan membuat sistem yang bertujuan untuk mengamankan data dalam aplikasi [1]. Kunci yang hanya diketahui oleh pengirim dan penerima, dari kunci tersebut bisa digunakan untuk mengembalikan ciphertext ke plaintext kembali oleh penerima [2]. Tujuan utama ilmu kriptografi ini juga berkaitan dengan keamanan informasi. Kerahasiaan digunakan untuk memastikan bahwa siapa pun yang memiliki otoritas atau kunci rahasia tidak dapat membuka atau mengupas data yang telah disandi [3].

Pada dasarnya, untuk mengurangi kelemahan dalam pengamanan data saat pengiriman file, algoritma kriptografi merusak atau menyembunyikan data, sehingga file yang akan dikirim ke penerima harus terlebih dahulu dienkripsi menggunakan algoritma kriptografi seperti AES, DES, RSA, Rijndael, Block Cipher, dan lainnya [4]. AES akan menjadi standar utama untuk mengenkripsi semua jenis data elektronik, termasuk yang digunakan dalam aplikasi komersial seperti telekomunikasi, informasi pribadi, perbankan dan finansial, serta transaksi finansial [5]. Enkripsi adalah proses mengubah data plaintext, yang merupakan data informasi yang dapat dibaca, menjadi data ciphertext, yang merupakan data acak, atau pesan yang tidak dapat dibaca [6]. Konsep enkripsi dan dekripsi digunakan untuk mengubah data atau pesan menjadi PlainText. Proses ini dikenal sebagai enkripsi dengan kunci [7].

AES adalah algoritma kriptografi berjenis cipher blok yang terkenal luas untuk pengenkripsi data [8]. Algoritma AES dimulai dengan proses pembuatan kunci. Pengguna dapat menetapkan panjang kunci dalam byte dengan mengatur parameter. Metode Kunci yang telah dibuat kemudian akan digunakan untuk mengenkripsi data

pengamanan data AES menggunakan bahasa dan library PyCryptodome pemrograman python dan penerapan [9]. AES memiliki panjang kunci variabel 128,192, dan 256 bit, memberikan keamanan dan kecepatan. AES menggunakan cipher blok. simetris dengan sepuluh putaran untuk kunci 128-bit, dua belas putaran untuk kunci 192-bit, dan empat belas putaran untuk kunci 144-bit, mengubah kunci 256-bit [10]. Jurnal ini bertujuan untuk memberikan penjelasan mendalam tentang cara kerja AES, salah satu algoritma enkripsi simetris yang paling aman dan banyak digunakan di dunia. Jurnal ini berfokus pada cara AES digunakan untuk mengamankan data dengan menggunakan Java. Jurnal ini juga bertujuan untuk mengevaluasi kinerja, efisiensi, dan keamanan penerapan algoritma AES di Java NetBeans, untuk memberikan contoh implementasi praktis, dan untuk mengidentifikasi tantangan yang mungkin dihadapi dalam proses pengembangan. Penelitian ini bertujuan untuk membantu memahami kinerja dan keandalan algoritma AES dalam melindungi data dalam berbagai kondisi dan memberikan saran untuk pengembangan lebih lanjut.

2. TINJAUAN PUSTAKA

H. Saputra Djong dan S. Siswanto menunjukkan bahwa menggunakan kombinasi teknik RC4 dan AES-256 dapat meningkatkan keamanan dokumen. Kombinasi ini melindungi data bisnis dengan baik [1].

A. Eka Putri dkk.,menunjukkan bahwa metode EOF dan algoritma AES 128 bit berhasil menyembunyikan dan mengamankan data dalam aplikasi berbasis Java. Metode ini dapat diandalkan untuk data sensitif [2].

Penelitian oleh S. Asri dkk, menunjukkan bahwa algoritma AES sangat efektif dan aman untuk melindungi data, dan mereka memimpin dalam kecepatan dan keamanan komunikasi jaringan [3].

C. Irawan dan A. Winarno, menemukan bahwa penggabungan algoritma AES dan DES meningkatkan enkripsi data daripada menggunakan satu algoritma. Kombinasi algoritma ini meningkatkan keamanan file dokumen [4].

R. Siringoringo, menunjukkan bahwa AES dan RSA keduanya dapat mengamankan file dengan baik, tetapi keduanya bekerja dengan cara yang berbeda. AES bekerja lebih cepat, sedangkan RSA memberikan keamanan tambahan [5].

F. O. Dayera dan M. B. Palungan, menunjukkan bahwa kriptografi dapat melindungi kerahasiaan dan integritas data [6].

A. A. I. Ramadhan dkk., menunjukkan bahwa AES di Java memungkinkan enkripsi dan dekripsi yang kuat, meningkatkan keamanan data. AES memastikan perlindungan data yang baik [7].

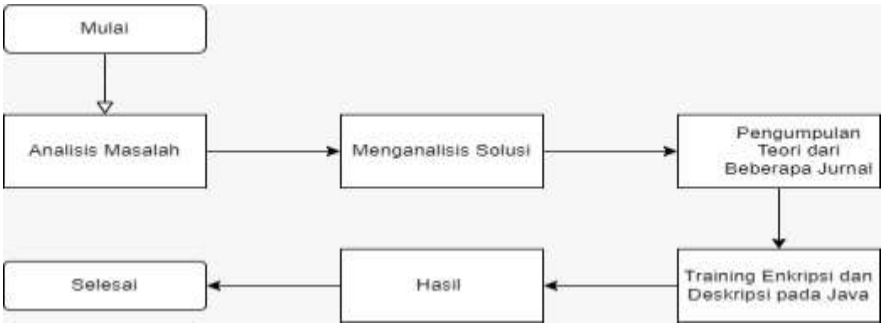
G. Y. P. Engko dkk., menemukan bahwa enkripsi ganda menggunakan AES dan RC5 memiliki keamanan yang lebih baik daripada enkripsi menggunakan satu algoritma. Ini menunjukkan bahwa enkripsi ganda sangat efektif untuk melindungi data [8].

Menurut A. M. Fajrin dkk., menyebutkan AES lebih cepat, tetapi RSA lebih aman dalam manajemen kunci, tergantung pada kecepatan atau keamanan [9].

Menurut M. Suwarni dkk., hasil penelitiannya menunjukkan bahwa AES lebih unggul dalam keamanan dan efisiensi dibandingkan DES. Ini karena AES memiliki perlindungan yang lebih kuat dan kecepatan enkripsi yang lebih tinggi, yang membuatnya lebih baik untuk mengamankan file [10].

Dari tinjuan pustaka penelitian terdahulu dapat disimpulkan bahwa Advanced Encryption Standard (AES) dapat melindungi data di semua sektor dan kompatibel dengan berbagai platform. Keterkaitan antara tinjaun pustaka dengan penelitian adalah sama-sama menggunakan Algoritma AES sebagai keamanan untuk melindungi data.

3. METODE PENELITIAN



Gambar 1. Alur metode penelitian

Pada gambar 1, merupakan gambar flowchart atau alur penelitian, berikut uraian rincian tahapan pada penelitian :

1. Analisis Masalah

Mencari, mengamati, dan memperhatikan masalah yang sedang terjadi dengan mengidentifikasi solusinya.

2. Menganalisis Solusi

Berdasarkan pengamatan masalah, menentukan solusi yang dapat digunakan untuk menyelesaikan masalah. Dalam penyelesaian masalah ini digunakan Java Net Beans untuk menyelesaikan masalah.

3. Pengumpulan teori dari beberapa jurnal

Dengan melakukan pengumpulan teori pendukung dari beberapa jurnal yang terkait dengan menggunakan studi literatur (mengumpulkan teori dari beberapa referensi jurnal).

4. Training enkripsi dan deskripsi teks pada Java

Membuat coding yang dapat meneksripsi dan deskripsi teks pada Java Net Beans dengan menggunakan algoritma AES 256.

5. Hasil

Hasil coding merupakan teks yang telah dienkripsi dan kemudian dideskripsikan kembali.

4. HASIL DAN PEMBAHASAN

4.1. Algoritma AES (Advanced Encryption Standard)

AES adalah metode enkripsi yang paling umum digunakan karena sederhana dan efektif. AES adalah blok cypher asimetrik yang menggunakan kunci yang sama untuk enkripsi dan dekripsi. AES dibuat untuk mengatasi kekurangan algoritma kriptografi yang sudah ada, seperti DES, yang memiliki kunci yang pendek dan kekurangan hardware. Algoritma AES dimulai dengan pembuatan tombol. Pengguna dapat menggunakan parameter untuk menentukan panjang kunci dalam byte. Kunci yang telah dibuat akan digunakan untuk mengenkripsi data. Untuk menjalankan proses pengamanan data AES, library PyCryptodome dan bahasa pemrograman python digunakan.

Blok Data: AES mengenkripsi data dalam unit blok 128 bit (16 byte). Namun, jika ukuran data yang ingin dienkripsi lebih besar dari unit blok tersebut, data tersebut dipecah menjadi beberapa blok. Kunci Enkripsi: AES menggunakan kunci dengan panjang 128, 192, atau 256 bit, dan kunci yang lebih panjang memiliki tingkat keamanan yang lebih tinggi. Untuk proses enkripsi dan dekripsi, kunci ini harus dirahasiakan.

Proses Enkripsi: SubBytes: Menggunakan tabel substitusi yang disebut S-Box untuk mengganti setiap byte data dengan byte yang berbeda. ShiftRows: Menggeser baris dalam blok data untuk mendistribusikan byte secara merata. MixColumns: Mencampur kolom data untuk meningkatkan keacakan data. AddRoundKey: Menggunakan kunci enkripsi untuk mengubah data pada setiap putaran.

Tabel 1. Round

Tipe	Jumlah Key (Nk)	Besar Blok (Nb)	Jumlah Round (Nr)
AES - 128	4	4	10
AES - 192	6	4	12
AES - 256	8	4	14

Tabel 1 merupakan tabel Putaran (Rounds): AES melibatkan beberapa putaran (rounds) tergantung pada panjang kunci. Round merupakan salah satu komponen penting dalam algoritma AES, round berfungsi untuk menjamin bahwa informasi dienkripsi dengan kuat dan sulit untuk didekripsi tanpa kunci yang tepat. Setiap putaran memberikan lapisan perlindungan tambahan untuk melindungi berbagai jenis serangan kriptografi.

4.2. Proses Kriptografi dalam Java

Gambar 2. Proses AES

Pada gambar 2 merupakan proses AES, memiliki Spesifikasi sebagai berikut:

a. Plain text : Pesan atau data yang dapat dibaca dalam bentuk aslinya.

b. Secret Key : Kunci yang digunakan untuk enkripsi dan dekripsi.

c. Cipher text: Pesan atau data yang sudah dienkripsi.

d. Cipher: Kelas ini merupakan inti dari kerangka Java Cryptographic Extension (JCE) karena menyediakan fitur cipher kriptografi untuk enkripsi dan dekripsi.

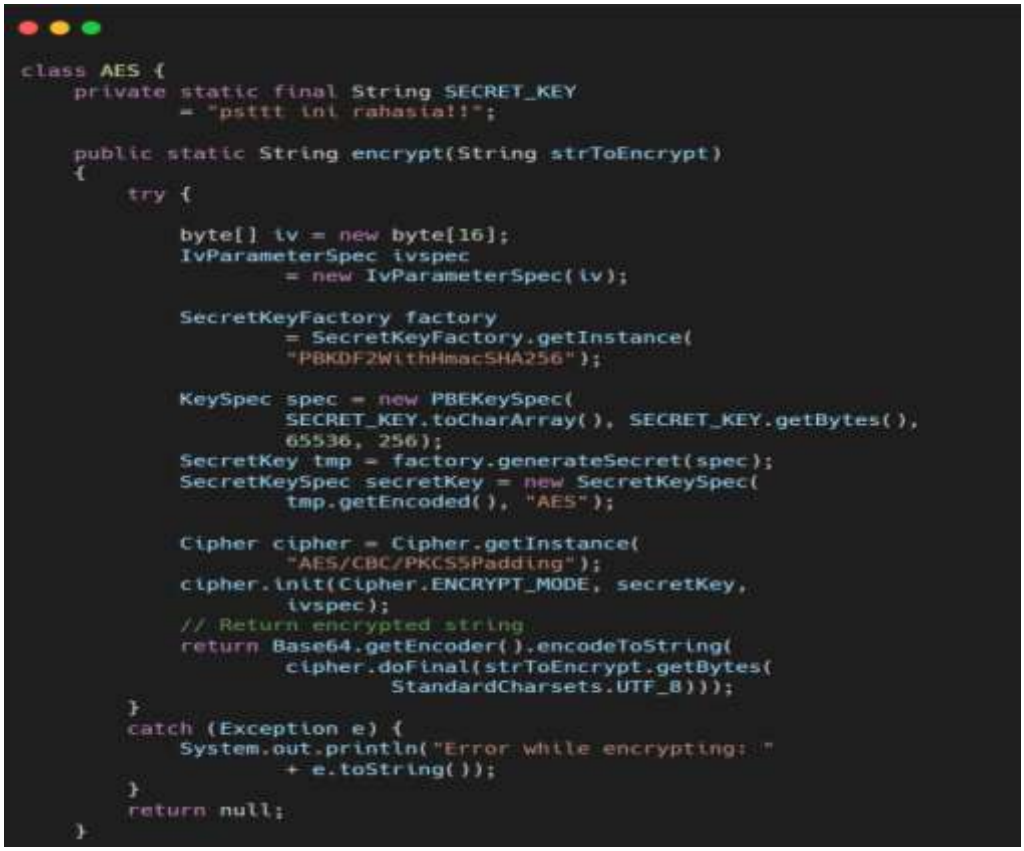
4.3. Proses Kriptografi dalam Java

Proses praktikum menggunakan Java pada NetBeans. Dengan membuat 2 class, class pertama yaitu AES dan Main.

```
package enkripsi;

import java.nio.charset.StandardCharsets;
import java.security.spec.KeySpec;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
```

Gambar 3. Class AES

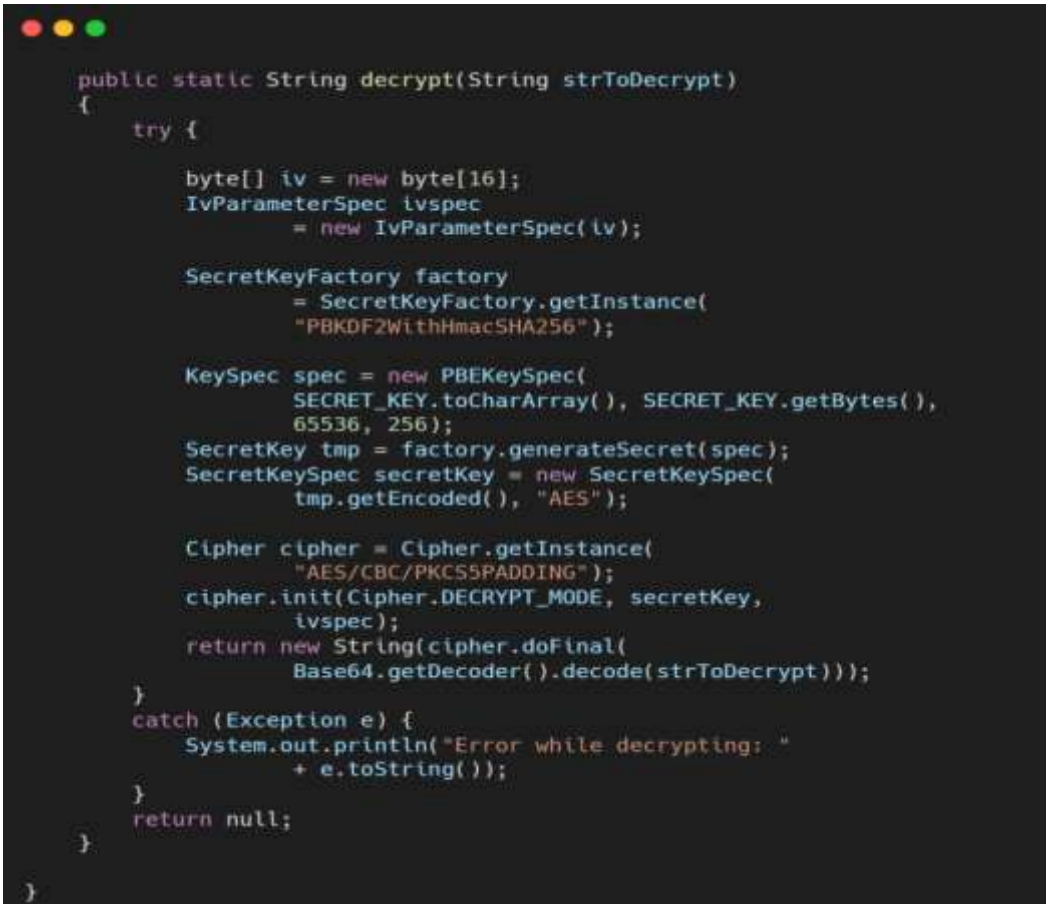
Pada gambar 3, merupakan class AES, Kelas yang dibutuhkan untuk operasi enkripsi dan deskripsi pada package enkripsi yang menggunakan bahasa pemrograman Java.



```
class AES {
    private static final String SECRET_KEY
            = "psttt ini rahasia!!";

    public static String encrypt(String strToEncrypt)
    {
        try {

            byte[] iv = new byte[16];
            IvParameterSpec ivspec
                    = new IvParameterSpec(iv);

            SecretKeyFactory factory
                    = SecretKeyFactory.getInstance(
                    "PBKDF2WithHmacSHA256");

            KeySpec spec = new PBEKeySpec(
                    SECRET_KEY.toCharArray(), SECRET_KEY.getBytes(),
                    65536, 256);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(
                    tmp.getEncoded(), "AES");

            Cipher cipher = Cipher.getInstance(
                    "AES/CBC/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey,
                    ivspec);
            // Return encrypted string
            return Base64.getEncoder().encodeToString(
                    cipher.doFinal(strToEncrypt.getBytes(
                            StandardCharsets.UTF_8)));
        }
        catch (Exception e) {
            System.out.println("Error while encrypting: "
                    + e.toString());
        }
        return null;
    }
```

Gambar 4. Class AES

Pada gambar 4 merupakan gambar class AES, PBEKeySpec Menghasilkan kunci AES dari password dengan iterasi dan panjang kunci yang ditentukan. Cipher Kelas Cipher disiapkan untuk mode AES/CBC/PKCS5Padding untuk enkripsi dan dekripsi. Enkripsi Menggunakan cipher.doFinal untuk mengenkripsi pesan dan mengubah hasilnya menjadi format Base64. Dekripsi Menggunakan cipher.doFinal untuk mendekripsi pesan yang telah dienkripsi dan mengembalikannya ke format aslinya.



```
    public static String decrypt(String strToDecrypt)
    {
        try {

            byte[] iv = new byte[16];
            IvParameterSpec ivspec
                    = new IvParameterSpec(iv);

            SecretKeyFactory factory
                    = SecretKeyFactory.getInstance(
                    "PBKDF2WithHmacSHA256");

            KeySpec spec = new PBEKeySpec(
                    SECRET_KEY.toCharArray(), SECRET_KEY.getBytes(),
                    65536, 256);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(
                    tmp.getEncoded(), "AES");

            Cipher cipher = Cipher.getInstance(
                    "AES/CBC/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey,
                    ivspec);
            return new String(cipher.doFinal(
                    Base64.getDecoder().decode(strToDecrypt)));
        }
        catch (Exception e) {
            System.out.println("Error while decrypting: "
                    + e.toString());
        }
        return null;
    }

}
```

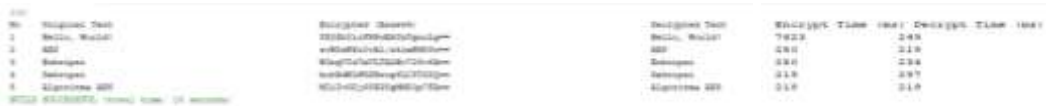
Gambar 5. String decryption

Pada gambar 5, merupakan string decryption. Kunci Rahasia Variabel SECRET_KEY adalah kunci rahasia yang digunakan untuk proses enkripsi dan dekripsi. Initialization Vector (IV) yaitu IvParameterSpec yang digunakan untuk mode CBC. IV panjangnya 16 byte. Secret Key Factory SecretKeyFactory digunakan untuk menghasilkan kunci dari password yang diberikan.

```
public class Main {
 public static void main(String[] args) {
        String[] originalTexts = {
            "Hello, World!",
            "AES",
            "Enkripsi",
            "Dekripsi",
            "Algoritma AES"
        };

        System.out.printf("%-5s %-50s %-60s %-50s %-15s %-15s%n", "No", "Original Text",
"Encrypted (Base64)", "Decrypted Text", "Encrypt Time (ms)", "Decrypt Time (ms)");

        for (int i = 0; i < originalTexts.length; i++) {
            String original = originalTexts[i];

            long startEncrypt = System.currentTimeMillis();
            String encrypted = encrypt(original);
            long endEncrypt = System.currentTimeMillis();

            long startDecrypt = System.currentTimeMillis();
            String decrypted = decrypt(encrypted);
            long endDecrypt = System.currentTimeMillis();

            long encryptTime = endEncrypt - startEncrypt;
            long decryptTime = endDecrypt - startDecrypt;

            System.out.printf("%-5d %-50s %-60s %-50s %-15d %-15d%n",
                    (i + 1), original, encrypted, decrypted, encryptTime, decryptTime);
        }
 }
}
```

Gambar 6. Class main

Program Java ini mengenkripsi dan mendekripsi tabel teks menggunakan algoritma tertentu, seperti AES. Seperti pada gambar 6 teks asli disimpan dalam array originalTexts dan diproses satu per satu dalam loop. Waktu pengkodean dan penguraian kode diukur menggunakan System.currentTimeMillis() untuk menghitung durasinya. Hasil proses dan waktu yang dibutuhkan dicetak dalam bentuk tabel menggunakan System.out.printf(). Program ini memungkinkan Anda mengevaluasi efisiensi enkripsi dan dekripsi dengan menampilkan hasil serta durasi prosesnya.



Gambar 7. Output atau hasil codingan

Gambar 7 adalah hasil dari program Java yang menunjukkan kinerja enkripsi dan dekripsi teks tertentu menggunakan algoritma enkripsi, kemungkinan AES. Setiap teks dalam array originalTexts diproses secara bergantian dan hasilnya dicetak dalam format tabel yang mencakup teks asli, hasil pengkodean Base64, hasil decoding, dan waktu yang diperlukan untuk proses pengkodean dan decoding dalam milidetik.

5. KESIMPULAN

Pada penelitian ini dapat disimpulkan bahwa Algoritma Advanced Encryption Standard (AES) digunakan untuk mengenkripsi dan mendekripsi pesan dengan menggunakan bahasa pemrograman Java. Algoritma Advanced Encryption sebagai algoritma kriptografi kunci simetris, menggunakan kunci yang sama untuk kedua proses enkripsi dan dekripsi, memastikan bahwa hanya orang yang memiliki kunci tersebut yang dapat mengakses informasi yang sebenarnya. Algoritma enkripsi yang digunakan AES berhasil mengenkripsi dan mendekripsi berbagai teks dengan akurasi yang lengkap, karena semua teks terenkripsi berhasil didekripsi dalam bentuk aslinya. Namun, efisiensi waktu pengkodean dan decoding berbeda, dengan pengkodean membutuhkan waktu lebih lama daripada decoding, seperti yang ditunjukkan pada teks "Hello World!" membutuhkan waktu pengkodean terlama (7623 milidetik). Variasi durasi ini mungkin disebabkan oleh kompleksitas atau biaya penerapan enkripsi. Namun, waktu decoding relatif stabil dan lebih cepat dibandingkan coding. Secara keseluruhan, program ini efektif dalam mengukur kinerja enkripsi dan dekripsi tetapi berpotensi meningkatkan efisiensi selama enkripsi.

DAFTAR PUSTAKA

[1] H. Saputra Djong and S. Siswanto, "Implementasi Kriptografi Dengan Menggunakan Metode Rc4 Dan Aes-256 Untuk Mengamankan File Dokumen Pada Pt Varnion Technology Semesta," *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, no. September, pp. 149–158, 2022.

[2] A. Eka Putri, A. Kartikadewi, and L. A. Abdul Rosyid, "Implementasi Kriptografi dengan Algoritma Advanced Encryption Standard (AES) 128 Bit dan Steganografi menggunakan Metode End of File (EOF) Berbasis Java Desktop pada Dinas Pendidikan Kabupaten Tangerang," *Appl. Inf. Syst. Manag.*, vol. 3, no. 2, pp. 69–78, 2021, doi: 10.15408/aism.v3i2.14722.

[3] S. Asri, T. Peryanto, E. Malays, and U. Yai, "Perbandingan Implementasi Algoritma Aes Dalam Pemrograman Dan Analisis Algoritma Enkripsi Untuk Pengamanan Komunikasi Jaringan," *J. Ilm. Tek. Inform.*, vol. 25, no. 1, pp. 80–87, 2024, [Online]. Available: https://doi.org/10.37817/tekinfo.v25i1

[4] C. Irawan and A. Winarno, "Kombinasi Algoritma Kriptogarafi Aes Dan Des Untuk Enkripsi File Dokumen Proposal," *Sendiu*, pp. 2–8, 2020.

[5] R. Siringoringo, "Analisis dan Implementasi Algoritma Rijndael (AES) dan Kriptografi RSA pada Pengamanan File," *KAKIFIKOM (Kumpulan Artik. Karya Ilm. Fak. Ilmu Komputer)*, vol. 02, no. 01, pp. 31–42, 2020, doi: 10.54367/kakifikom.v2i1.666.

[6] F. O. Dayera, Musa Bundaris Palungan, "G-Tech : Jurnal Teknologi Terapan," *G-Tech J. Teknol. Terap.*, vol. 8, no. 1, pp. 186–195, 2024, [Online]. Available: https://ejournal.uniramalang.ac.id/index.php/g-tech/article/view/1823/1229

[7] A. A. I. Ramadhan, E. Z. Rivanti, and R. S. Zulva, "Implementasi Kriptografi AES Menggunakan Bahasa Java Programming: Meningkatkan Keamanan Data Melalui Enkripsi & Dekripsi Yang Kuat," *J. Pendidik. Teknol. Inf.*, pp. 20–26, 2023, [Online]. Available: https://jurnal.umj.ac.id/index.php/TripleA/article/view/17513%0Ahttps://jurnal.

umj.ac.id/index.php/TripleA/article/download/17513/9646

[8] G. Y. P. Engko M, A. Id Hadiana, and P. Nurul Sabrina, "Kriptografi Untuk Enkripsi Ganda Pada Gambar Menggunakan Algoritma AES (Advanced Encryption Standard) Dan RC5 (Rivest Code 5)," *Informatics Digit. Expert*, vol. 4, no. 1, pp. 25–32, 2022, doi: 10.36423/index.v4i1.884.

[9] A. M. Fajrin, C. Kelvin, B. Owen, and B. Aji, "Perbandingan Performa dari Algoritma AES dan RSA dalam Keamanan Transaksi," vol. 5, no. 2, pp. 696–705, 2024.

[10] M. Suwarni, J. Wahyudi, and K. Khairil, "Comparison of the DES Cryptographic Algorithm and the AES Algorithm in Securing Document Files," *J. Media Comput. Sci.*, vol. 2, no. 1, pp. 41–48, 2023, doi: 10.37676/jmcs.v2i1.3348.