Jurnal Algoritma
E-ISSN: 2302-7339 | P-ISSN: 1412-3622
INSTITUT TEKNOLOGI GARUT

Implementasi Algoritma *Advanced Encryption Standard* untuk Pengamanan Data Pengguna Aplikasi Media Sosial VirCle

Mohammad Fajar[1], Audri Billy Kambodji[2], Izmy Alwiah Musdar[3]

Jurnal Algoritma
STMIK Kharisma Makassar
Jl. Baji Ateka No. 20 Makassar 90131, Sulawesi Selatan, Indonesia
Email : stmik@kharisma.ac.id

[1]fajar@kharisma.ac.id
[2]audribilly@gmail.com
[3]izmyalwiah@kharisma.ac.id

Abstrak – VirCle merupakan salah satu aplikasi media sosial berbasis *mobile* yang dikembangkan sebagai media komunikasi dan interaksi antara sesama penggunanya. Data pengguna aplikasi disimpan pada basis data *Firebase*, tanpa adanya mekanisme keamanan tambahan. Meskipun *Firebase* menyediakan layanan pengamanan ketika data ditrasmisi serta mekanisme pengaturan akses, akan tetapi dikarenakan *Firebase* merupakan layanan basis data *back-end,* dimana aplikasi pengakses dan datanya berada pada lapisan yang terpisah, maka data pengguna VirCle yang tersimpan tersebut masih dapat diakses, baik itu oleh aplikasi eksternal lain melalui antarmuka pemrograman, atau pengguna dengan peran tertentu di sistem seperti administrator. Di sisi lain, keberadaan data pengguna yang tersimpan tersebut merupakan hal yang sangat penting untuk senantiasa dijaga kerahasiaannya oleh pengembang dan pengelola layanan aplikasi VirCle. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan algoritma *Advanced Encryption Standard* (AES) untuk enkripsi data pengguna aplikasi VirCle sehingga memiliki tambahan lapisan keamanan. Rancangan enkripsi menggunakan algoritma AES-128bit dengan pertimbangan kecepatan proses enkripsi dan dekripsi yang relatif lebih cepat dibandingkan algoritma kriptografi simetris lainnya. Implementasi AES pada studi ini menggunakan bahasa Kotlin dengan memanfaatkan *library* AES 128-bit BouncyCastle dan mode operasi CBC dan *padding* PKCS5 untuk membantu proses enkripsi dan dekripsi. Hasil evaluasi menunjukkan aplikasi VirCle yang dikembangkan dapat melakukan enkripsi data pengguna (*plaintext*) menjadi data yang terkode (*chipertext*) untuk kemudian disimpan pada basis data *Firebase*, selanjutnya berhasil melakukan pembacaan dan dekripsi data *chipertext* yang tersimpan dari basis data untuk digunakan kembali. Evaluasi kinerja aplikasi dari aspek kecepatan menunjukkan waktu proses enkripsi dan dekripsi data pengguna yang cukup cepat dengan rata-rata total waktu sebesar 1.35 milidetik untuk proses enkripsi, dan 1.53 milidetik untuk proses dekripsinya. Selain itu, evaluasi *Avalanche Effect* (AE) dari hasil enkripsi algortima AES-128 bit yang digunakan, didapat rata-rata persentasi sebesar 50.36%, hal ini menunjukkan tingkat keacakan data dipandang baik.

Kata Kunci – *Advanced Encryption Standard*; Aplikasi Vircle; Enkripsi; Dekripsi.

I. PENDAHULUAN

Perkembangan teknologi informasi membawa perubahan yang besar pada kehidupan masyarakat. Salah satunya yaitu kebebasan personal dalam memberikan komentar, ide, saran dan kritik terhadap berita atau informasi tertentu yang tak jarang dijumpai setiap hari melalui berbagai media digital [1], [2], termasuk aplikasi media sosial yang merupakan media digital yang cukup banyak digunakan masyarakat hari ini untuk berkomunikasi dan berinteraksi [3]. Di sisi lain, pemakaian beragam aplikasi media sosial ini tentunya

memiliki dampak pada keamanan data dari penggunanya, di mana data pribadi pengguna menjadi rentan untuk dieksploitasi oleh pihak yang tidak bertanggung jawab, seperti pelaku *cybercrime* [4], [5], untuk selanjutnya digunakan dalam tindakan penipuan, pencurian atau penyebaran identitas, hingga melakukan pemerasan dan pelecehan secara *online*. Untuk mengatasi masalah kemanan data pada aplikasi-aplikasi media sosial ini, diperlukan sebuah sistem keamanan yang dapat mencegah terjadinya hal yang tidak diinginkan terhadap data pengguna [6], dengan tetap mengacu pada tiga prinsip utama keamanan sistem informasi yaitu kerahasiaan (*confidentiality)*, integritas (*integrity*), dan ketersediaan (*availability*) atau yang disingkat CIA [7].

Saat ini, peneliti mengembangkan aplikasi VirCle, sebuah aplikasi media sosial berbasis *mobile*. Serupa dengan aplikasi-aplikasi media sosial lainnya, VirCle dapat digunakan untuk berkomunikasi dan berinteraksi antara sesama penggunanya. Data pengguna aplikasi VirCle tersebut disimpan pada basis data *Firebase*, tanpa adanya mekanisme keamanan data tambahan. Meskipun *platform Firebase* menyediakan layanan pengamanan ketika data ditransmisi serta tersedianya mekanisme pengaturan akses, tetapi karena *Firebase* merupakan layanan basis data *back-end,* dimana aplikasi pengakses dan datanya berada pada lapisan yang terpisah, maka data pengguna VirCle yang tersimpan tersebut masih dapat diakses, baik itu oleh aplikasi eksternal lain melalui antarmuka pemrograman, atau pengguna dengan peran tertentu di sistem seperti administrator yang memiliki akses lebih luas. Sementara keberadaan data pengguna yang tersimpan tersebut merupakan hal yang sangat penting untuk senantiasa dijaga kerahasiaannya oleh pengembang dan pengelola layanan aplikasi VirCle. Untuk keperluan sistem kemanan data pengguna, maka dalam penelitian ini, dipilih teknik kriptografi dengan pertimbangan fokus pengamanan pada data pelanggan yang akan disimpan di basis data *Firebase.* Teknik kriptografi merupakan ilmu yang mempelajari metode matematis yang berkaitan dengan informasi dan tujuan keamanan informasi, seperti kebenaran informasi, kualitas dan keamanan informasi [8]. Algoritma kriptografi yang digunakan yaitu *Advanced Encryption Standard (*AES*)* merupakan algoritma kriptografi simetris yang paling umum digunakan untuk mengenkripsi dan mendekripsi data, dengan keunggulan kecepatan prosesnya relatif lebih cepat dibandingkan dengan algoritma kriptografi simetris lainnya [9].

Sejumlah penelitian terkait pemanfaatan Algoritma AES telah dikaji dalam literatur. Diantaranya, oleh Halim dan Sugiarto [3] dimana dikembangkan sebuah aplikasi media sosial bernama WeConnect dengan memanfaatkan Algoritma AES untuk mengenkripsi *chat* yang dikirimkan pengguna. Studi ini tidak melakukan pengukuran performa algoritma enkripsi atau aplikasi yang menggunakan algoritma AES. Demikian pula studi yang dilakukan oleh Laila Mustaka [4] yang menerapkan enkripsi pada proses *login* mengunakan algoritma AES serta steganografi menggunakan LBS. Hasil penelitian ini menunjukkan bahwa algoritma berhasil diterapkan pada proses *login*, namun tidak memaparkan tingkat keamanan proses *login* setelah penerapan algoritma enkripsi. Di sisi lain, Delisman dkk melakukan kajian tentang implementasi algoritma AES untuk keamanan file hasil radiologi [8]. Kunci enkripsi yang digunakan adalah *password* yang diinputkan pengguna, sehingga file tidak dapat didekripsi tanpa menginput *password* yang sesuai. Penelitian ini menunjukkan bahwa algoritma enkripsi AES dapat digunakan untuk enkripsi dan dekripsi file gambar. Meskipun sejumlah studi tersebut memperlihatkan bahwa algoritma AES berhasil diterapkan untuk proses enkripsi dan dekripsi data, akan tetapi evaluasi performa aplikasi ataupun algoritma AES yang digunakan belum dilakukan. Oleh karena itu, dalam penelitian ini, evaluasi tidak hanya dilakukan terhadap keberhasilan enkripsi dan dekripsi data pengguna, namun juga terhadap kecepatan proses enkripsi dan dekripsi data, serta kualitas enkripsi dengan menghitung tingkat keacakan hasil enkripsinya. Pemilihan AES-128 bit pada studi ini didasarkan dengan pertimbangan bahwa blok 128 bit cukup baik untuk memberikan mekanisme tambahan pengamanan data, sekaligus membutuhkan sumber daya yang lebih kecil dibandingkan blok AES-192 dan AES-256. Selain itu, berbeda dengan studi-studi sebelumnya, dalam penelitian ini digunakan mode operasi *chipper* blok (CBC) pada tahap inisialisasi kunci dengan tujuan untuk memperoleh kunci yang unik [10].

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk mengimplementasikan algoritma AES dalam proses enkripsi dan dekripsi data pengguna aplikasi media sosial VirCle, sehingga diharapkan dapat mengantisipasi potensi ancaman keamanan dan kebocoran data penggunanya. Hasil penelitian ini diharapkan memberikan kontribusi berupa masukan saran dan ide pada peneliti dan pengembang aplikasi media sosial yang memerlukan tambahan lapisan keamanan pada aplikasi media sosial.

II. METODOLOGI PENELITIAN

A. Tahapan Penelitian

Penelitian ini menggunakan alur pengembangan perangkat lunak yang dimulai dari studi literatur dengan merujuk pada studi atau penelitian terkait untuk mendapatkan informasi pemanfaatan algoritma AES termasuk kinerjanya, kemudian melakukan analisis kebutuhan sistem untuk enkripsi dan dekripsi, membuat desain arsitektur sistem, desain proses enkripsi dan dekripsi, dilanjutkan dengan mengimplementasikan hasil desain ke *platform* target menggunakan bahasa pemrograman Kotlin untuk sistem operasi Android. Langkah terakhir yaitu melakukan pengumpulan data melalui evaluasi sistem dalam dua tahap. Tahap pertama, evaluasi proses enkripsi dan dekripsi data serta waktu yang dibutuhkan untuk proses tersebut, dan tahap kedua yaitu evaluasi tingkat keacakan hasil enkripsi dengan menghitung *Avalanche Effect* (AE). Hasil evaluasi digunakan sebagai dasar dalam menyusun kesimpulan penelitian. Gambar 1 memperlihatkan tahapan-tahapan pada penelitian ini.

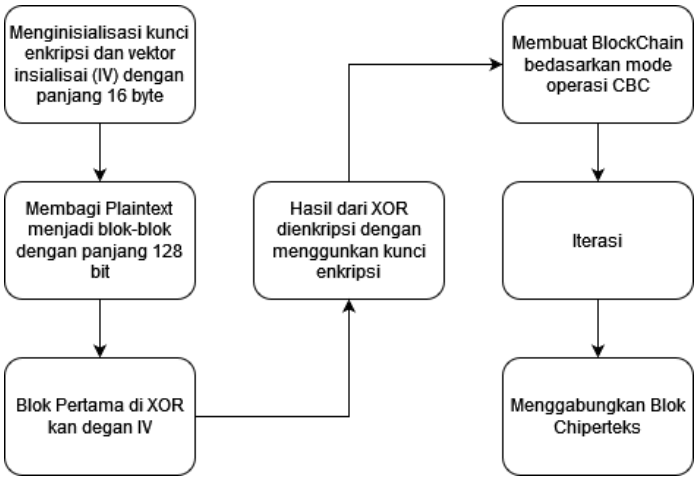
Gambar 1: Tahapan Penelitian

Dalam evaluasi proses enkripsi dan dekripsi, digunakan sebanyak 30 data pengguna sebagai ujicoba yang terdiri dari *username, full name, email,* dan *password*. Evaluasi ini untuk mengetahui apakah aplikasi yang telah dirancang berhasil melakukan proses enkripsi dan menyimpan data pengguna hasil enkripsi tersebut ke basis data *Firebase*, serta apakah proses dekripsi dapat dilakukan untuk menampilkan atau menggunakan kembali data aslinya. Untuk evaluasi waktu proses, penghitung waktu disisipkan pada kode implementasi, sementara evaluasi tingkat *Avalance Effect* menggunakan tools Crytools 2.1 untuk melakukan simulasi enkripsi dari algoritma AES yang digunakan.

B. Desain Enkripsi dan Dekripsi

Proses enkripsi dan dekripsi data yang dilakukan berdasarkan algoritma AES menggunakan mode operasi CBC dan *padding* PKCS5 untuk membantu proses tersebut. Algoritma AES yang diimplementasikan memiliki panjang kunci 128-bit. Berikut mekanisme proses Enkripsi dan Dekripsi algoritma AES 128-bit dengan mode operasi CBC dan *Padding* PKCS5 pada aplikasi VirCle:

1. Proses Enkripsi



Gambar 2: Proses Enkripsi Algoritma AES 128-bit

Gambar 2 menunjukkan proses enkripsi dari algoritma AES-128 bit dengan menggunakan mode

operasi CBC dan *padding* PKCS5. Proses ini cukup berbeda dengan proses enkripsi algoritma AES pada umumnya. Berikut penjelasan tahapan prosesnya:

a. Proses yang pertama yaitu menginisialisasi kunci enkripsi dan Vektor Inisialisasi (IV). Vektor Inisialisasi digunakan pada mode operasi CBC untuk di XOR-kan dengan blok pertama dari *plaintext* agar hasil enkripsi lebih rumit [10], [11]. Kunci enkripsi dan IV harus memiliki panjang 128 bit dan unik karena kelemahan dari algortima AES adalah kunci enkripsinya. Implementasinya sebagai berikut :
   - Menambahkan 'Bouncy Castle Provider' dari *library* bernama 'BouncyCastle' :
     `Security.addProvider(BouncyCastleProvide())`
   - Menginisialisasi kunci enkripsi dan objeknya
     ```
     val encryptionKey = "1234567890123456" //16-byte key
     val secretKey = SecretKeySpec(encryptionKey.toByteArray())
     ```
   - Menginisialisasi Vektor Inisialisasi (IV) dan Objeknya
     ```
     val iv = "abcdefghijklmnop" //16-byte IV
     val ivParameterSpec = IvParameterSpec(iv.toByteArray())
     ```
   - Membuat objek dari *chiper* dengan mode CBC dan *padding* PKCS5.
     `val cipher = Cipher.getInstance("AES/CBC/PKCS5Padding", "BC")`

b. Setelah menginisialisasi kunci enkripsi dan IV, *plaintext* akan dibagi menjadi beberapa blok dengan panjang 128 bit. Sebagai contoh terdapat data nama pengguna yaitu "Vinny Hong".
   Blok Pertama: Vinny Hong (9 byte)
   Karena panjang dari blok pertama kurang dari 128 bit atau 16 byte, dan didapatkan 16 – 9 byte = 7 byte, sehingga masih membutuhkan 7-byte lagi. *Padding* PKCS5 berfungsi untuk menambah panjang blok yang didapatkan dari *plaintext.*
   Panjang blok *plaintext: 9 bytes*
   Jumlah *byte padding* yang diperlukan, yaitu : 16 - 9 = 7 byte
   Byte *padding* yang diperlukan: 07 07 07 07 07 07 07 (dalam heksadesimal)
   Blok *plaintext* setelah *padding*, yaitu : Vinny hong 07 07 07 07 07 07 07

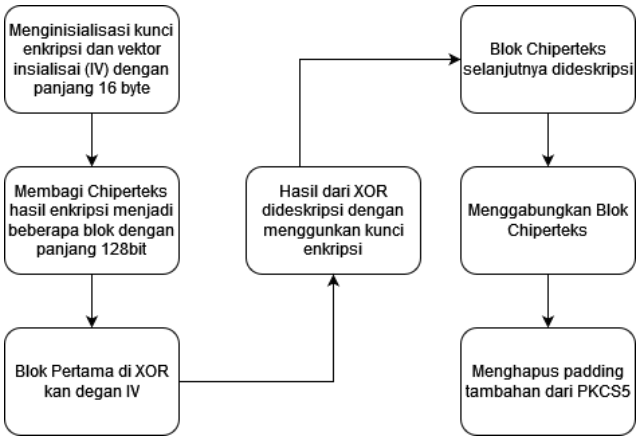
c. Proses selanjutnya adalah blok pertama dari *plaintext* akan di *X*OR kan dengan IV. Proses ini berasal dari mode operasi CBC dan tidak terdapat pada proses enkripsi algoritma AES secara umum.
   Blok pertama dan IV akan diubah menjadi bentuk biner.
   Blok Pertama: 01010110 01101001 01101110 01101110 01111001 00100000 01101000 01101111 01101110 01100111 00100000 00110000 00110111 00100000 00110000 00110111 00100000 00110000 00110111 00100000 00110000 00110111 00100000 00110000 00110111 00100000 00110000 00110111 00100000 00110000 00110111 00100000 00110000 00110111
   IV: 00110000 00110001 00110010 00110011 00110100 00110101 00110110 00110111 00111000 00111001 01000001 01000010 01000011 01000100 01000101 01000110
   Dan Hasil XOR dari kedua bilangan biner tersebut adalah:
   01100110 01011000 01000110 01000100 01001101 01011100 01100110 01001010 01011010 01001100 01101100 01101101 01101100 01101100 01100110 01101110 01101100

d. Hasil dari XOR tersebut akan dienkripsi dengan kunci enkripsi yang telah diinisialisasi sebelumnya, dan sesuai dengan proses umum dari proses enkripsi algoritma AES. Di mana hasil XOR tersebut akan melalui proses transformasi yaitu *Addroundkey*, *Subbyte*, *shiftrows*, dan *mixcoloms* [12]–[16] sebanyak 10 putaran (*round)* sesuai dengan panjang kunci dari algoritma AES yang digunakan [17].
   Berikut penjelasan singkat dari proses yang dilalui:
   Putaran Pertama:
   AddRoundKey: Blok hasil XOR di-XOR-kan dengan kunci enkripsi putaran pertama.
   SubBytes: Setiap byte disubstitusi dengan nilai dari tabel S-Box.
   ShiftRows: Baris kedua digeser satu byte ke kiri, baris ketiga digeser dua byte ke kiri, dan baris keempat digeser tiga byte ke kiri.
   MixColumns: Setiap kolom diubah dengan operasi linier yang melibatkan perkalian matriks.
   Putaran kedua hingga kesepuluh:
   Putaran yang sama seperti putaran pertama dilakukan, tetapi tanpa langkah *MixColumns* di

putaran terakhir.

e. Proses berikutnya adalah pembentukan rantai, proses ini berasal dari mode operasi CBC di mana hasil enkripsi akan disusun seperti rantai. Hasil dari enkripsi blok pertama akan menjadi blok *chiper text* pertama.
f. Proses berikutnya adalah proses iterasi dimana blok *chipertext* pertama pada *blockchain* akan di XOR kan dengan blok *plaintext* kedua. *Be*rikut penjelasan prosesnya:
   Hasil XOR blok *plaintext* pertama dengan IV akan dienkripsi dengan kunci enkripsi yang telah diinisialisasi, dan hasil dari enkripsi pertama akan menjadi blok *chipertext* pertama pada *blockchain*.
   Blok *chipertext* pertama yang dihasilkan akan di XOR kan dengan blok *plaintext kedua.*
   Hasil dari XOR tersebut akan dienkripsi dengan kunci enkripsi.
   Dan hasil dari enkripsi tersebut akan menjadi blok *chipertext* kedua.
   Proses akan berjalan berulang sampai blok *plaintext habis.*
g. *Se*telah semua blok *plaintext telah* dienkripsi maka semua blok *chipertext* akan digabungkan dan didapatkan hasil enkripsi dari *plaintext.*

2. Proses Dekripsi



Gambar 3: Proses Dekripsi Algoritma AES

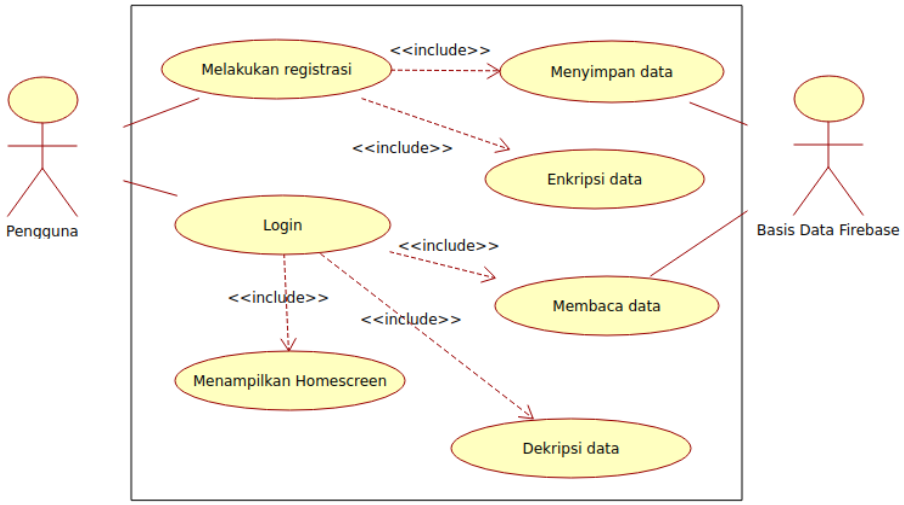
Pada Gambar 3 disajikan proses dekripsi dari algoritma AES-128 bit menggunakan mode operasi CBC dan *padding* PKCS5. Proses ini cukup berbeda dengan proses dekripsi algoritma AES pada umumnya. Berikut Penjelasan tahapan prosesnya:

1) Proses yang pertama adalah menginisialisasi kunci enkripsi dan Vektor Inisialisasi (IV). Kunci enkripsi dan IV harus memiliki panjang 128bit dan unik [10], [11], dan harus sama dengan yang digunakan pada proses enkripsi.
2) Setelah menginisialisasi kunci enkripsi dan IV, *Chipertext* hasil enkripsi sebelumnya akan dibagi menjadi beberapa blok dengan panjang 128 bit. Proses ini sama dengan proses pembagian dari *plaintext* pada proses enkripsi.
3) Blok *chipertext* pertama akan di XOR kan IV sama seperti proses enkripsi sebelumnya.
4) Hasil dari XOR tersebut didekripsikan dengan kunci enkripsi dimana akan melalui proses transformasi yaitu *AddRoundKey* (XOR dengan kunci putaran terkait), Inv*MixColumns* (operasi matriks terbalik), Inv*ShiftRows* (geser baris terbalik), dan Inv*SubBytes* (substitusi byte terbalik) [14], [16]. Proses tersebut berulang sebanyak 10 putaran (*Round)*.
5) Setelah blok *chipertext* pertama didekripsi, blok *chipertext* selanjutnya akan melalui proses yang sama hanya saja tidak perlu melakukan XOR dengan IV.
6) Dan setelah semua blok *chipertext* didekripsi, blok-blok tersebut akan digabungkan dan jika terdapat blok *padding* yang ditambah dengan PKCS5 akan dihilangkan.
7) Setelah blok *padding* dihilangkan maka didapatkan hasil dekripsi dari *plaintext yang di* enkripsi sebelumnya.

III. HASIL DAN PEMBAHASAN

A. Analisis Sistem

Pada awal pengembangan, aplikasi VirCle telah memiliki fitur registrasi, *login*, obrolan seperti mengirim pesan dan membaca pesan antar sesama pengguna yang terdaftar, serta *logout*. Untuk fitur enkripsi yang diusulkan ditambahkan pada aktifitas Melakukan registrasi, sebelum data registrasi pengguna disimpan ke basis data *Firebase*, maka terlebih dahulu dilakukan proses enkripsi. Untuk fitur dekripsi ditambahkan pada aktifitas menampilkan data pengguna ketika tampilan *Home screen* dijalankan, setelah pengguna berhasil *login* ke sistem. Proses enkripsi dan dekripsi data dipicu oleh pengguna melalui fitur Melakukan registrasi dan *Login*. Gambar 1 menunjukkan potongan spesifikasi kebutuhan fungsional Enkripsi dan Dekripsi Data Pengguna menggunakan diagram *use case* pada aplikasi VirCle.

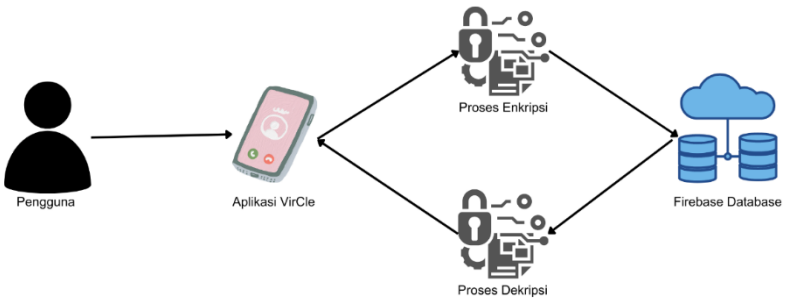


Gambar 1: Diagram *Usecase* Enkripsi dan Deskripsi pada Aplikasi VirCle

B. Arsitektur Sistem

Arsitekur sistem dibuat dengan tujuan untuk menggambarkan komponen utama sistem aplikasi VirCle terhadap proses enkripsi dan dekripsi data pengguna. Komponen sistem secara umum terdiri dari pengguna, Aplikasi sosial media VirCle yang berjalan diatas *platform Mobile* berbasis Android, proses Enkripsi dan proses Dekripsi, serta basis data *Firebase*.

Pengguna melakukan interaksi pada aplikasi VirCle yaitu melakukan registrasi, lalu aplikasi tersebut melakukan proses enkripsi dengan menggunakan algoritma AES-128 bit terhadap data pengguna. Setelah proses enkripsi selesai, maka data hasil enkripsi (*chipertext*) dikirim dan disimpan ke basis data *Firebase.* Setelah itu data hasil enkripsi yang telah disimpan akan melalui proses dekripsi, apabila pengguna berhasil *login* dan antarmuka *Homescreen* aplikasi VirCle ditampilkan.



Gambar 2: Arsitektur Sistem Aplikasi VirCle

C. Implementasi

Implementasi aplikasi VirCle termasuk fitur enkripsi dan dekripsinya menggunakan bahasa Kotlin dengan *platform* target Android. Lingkungan pengembangan terpadu (IDE) yang digunakan yaitu Android Studio. Proses enkripsi dan dekripsi algoritma AES-128 bit memanfaatkan *library* 'BouncyCastle' dengan menggunakan mode operasi CBC dan *padding* PKCS5 [18]. Berikut uraian implementasi rancangan untuk proses enkripsi:

1. Menambahkan 'Bouncy Castle Provider':
   Security.addProvider(BouncyCastleProvider())
2. Menginisialisasi kunci enkripsi dan objeknya :
   val encryptionKey = "1234567890123456" // 16-byte key
   val secretKey = SecretKeySpec(encryptionKey.toByteArray(), "AES")
3. Menginisialisai Vektor Inisialisasi (IV) dan objeknya :
   val iv = "abcdefghijklmnop" // 16-byte IV
   val ivParameterSpec = IvParameterSpec(iv.toByteArray())
4. Selanjutnya membuat objek dari *chiper* dengan mode CBC dan *padding* PKCS5 :
   val cipher = Cipher.getInstance("AES/CBC/PKCS5Padding", "BC")
5. Menginisialisasi variabel untuk menyimpang hasil enkripsi :
   val encryptedName = encryptData (fire*Username*, cipher, secretKey, iv ParameterSpec)
   val encrypted*Fullname* = encryptData (fire*Fullname*, cipher, secretKey, iv ParameterSpec)
   val encrypted*Email* = encryptData (fire*Email*, cipher, secretKey, ivParameter Spec)
   val encrypted*Password* = encryptData (fire*Password*, cipher, secretKey, iv ParameterSpec)
6. Selanjutnya merupakan fungsi enkripsi dari *library* 'Bouncy Castle" :
   fun encryptData(data: String, cipher: Cipher, secretKey: SecretKeySpec,
   ivParameterSpec: IvParameterSpec): ByteArray {
   cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivParameterSpec)
   return cipher.doFinal(data.toByteArray())}

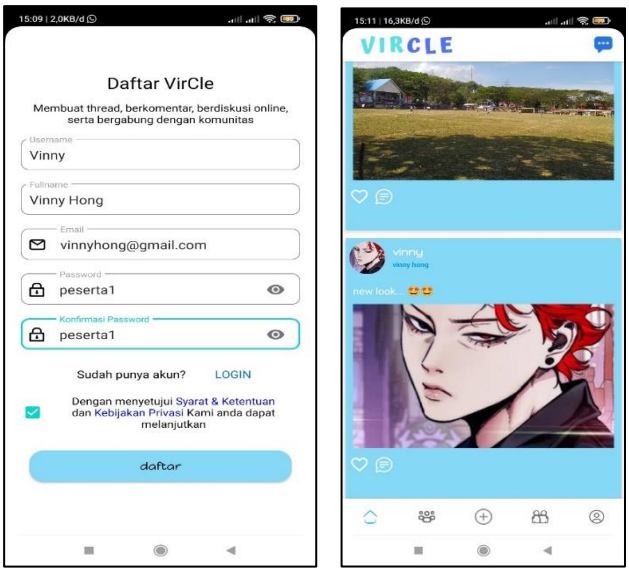
Untuk uraian proses dekripsinya disajikan sebagai berikut:

1. Menginisialisasi kunci enkripsi yang digunakan pada proses enkripsi :
   val encryptionKey = "1234567890123456" // 16-byte key
   val secretKey = SecretKeySpec(encryptionKey.toByteArray(), "AES")
2. Menginisialisai IV yang sama pada proses enkripsi :
   val iv = "abcdefghijklmnop" // 16-byte IV
   val ivParameterSpec = IvParameterSpec(iv.toByteArray())
3. Selanjutnya membuat objek dari *chiper* dengan mode CBC dan *padding* PKCS5 :
   val cipher = Cipher.getInstance("AES/CBC/PKCS5Padding", "BC")
4. Selanjutnya menginisialisasi variabel untuk menyimpang hasil dekripsi :
   val decrypted*Email* = decryptData(Base64.decode(user!!.get*Email*(), Base 64. DEFAULT), cipher, secretKey, ivParameterSpec)
   val decrypted*Fullname* = decryptData(Base64.decode(user!!.get*Fullname*(), Base64.DEFAULT), cipher, secretKey, ivParameterSpec)
   Valdecrypted*Username*=decryptData(Base64.decode(user!!.get*Username*(), Base64.DEFAULT), cipher, secretKey, ivParameterSpec)
   val decrypted*Password* = decryptData(Base64.decode(user!!.get*Password*(), Base64.DEFAULT), cipher, secretKey, ivParameterSpec)
5. Selanjutnya merupakan fungsi enkripsi dari *library* 'Bouncy Castle" :
   fun decryptData(data: ByteArray, cipher: Cipher, secretKey: SecretKeySpec,
   ivParameterSpec: IvParameterSpec): String {
   cipher.init(Cipher.DECRYPT_MODE, secretKey, ivParameterSpec)
   val decrypted = cipher.doFinal(data)
   return String(decrypted)}

D. Evaluasi

1. Enkripsi Dan Dekripsi

Evaluasi ini bertujuan untuk menguji apakah fitur enkripsi dan dekripsi data pengguna aplikasi berhasil dilakukan. Ujicoba dilakukan dengan menjalankan fitur daftar pengguna, kemudian memasukkan data registrasi (*plaintext*), selanjutnya tombol daftar ditekan untuk melakukan proses enkripsi data isian dan mengirim serta menyimpannya ke basis data *Firebase*. Gambar 1 memperlihatkan proses pengisian dan pendaftaran data pengguna aplikasi. Untuk ujicoba proses dekripsi, pertama-pertama dilakukan login, kemudian aplikasi melakukan pembacaan data terenkrip (*chipertext*) dari basis data Firbase, melakukan proses dekripsi, melakukan validasi pengguna, serta menampilkan *username* dan *full name* pengguna di *homescreen*. Gambar 2 menyajikan *homescreen* yang berhasil menampilkan data *username* dan *full name* pengguna setelah didekripsi.



(a) Proses Registrasi/Daftar (b) Menampilkan nama pengguna

Gambar 5: Ujicoba Fitur (a) Enkripsi data pada proses Registrasi/Daftar (b) Dekripsi data dengan menampilkan nama pengguna pada *Homescreen*

Untuk kebutuhan data uji coba, digunakan 30 data pengguna aplikasi VirCle. Tabel 1 memperlihatkan lima contoh data (*plaintext*) yang digunakan, sementara Tabel 2 menyajikan contoh data pengguna yang tersimpan di basis data merupakan hasil enkripsi (*chipertext*) yang berhasil dilakukan.

Tabel 1: Data Pengguna asli (Plaintext)

No.	***username***	***fullname***	***email***	***password***
1	Vinny	Vinny hong	vinnyhong@gmail.com	peserta1
2	Dom	Dom Kang	domination@gmail.com	peserta2
3	Super Rookie	Jay jo	Jayjo@gmail.com	peserta3
4	Minu	Minu yoon	minu@gmail.com	peserta4
5	Owen	Owen Knight	knight@gmail.com	peserta5

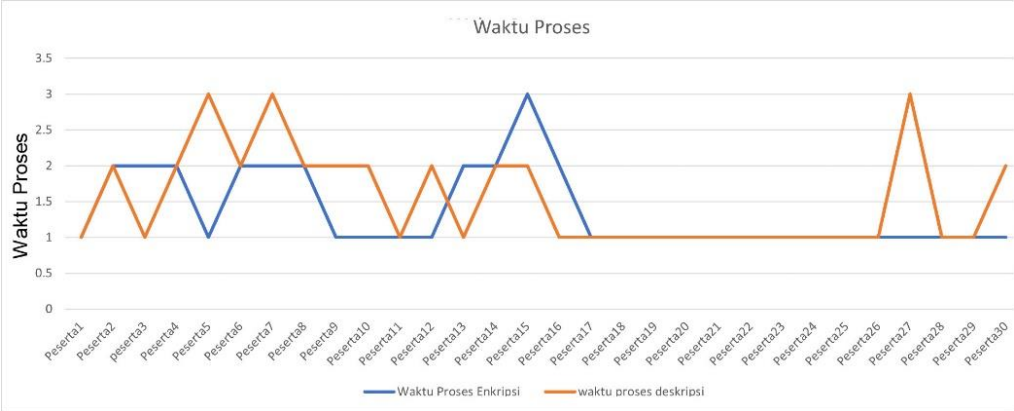
Tabel 2 : Data pengguna yang telah dienkripsi *(Chipertext)*

NO	Data Enkripsi			
	username	***fullname***	***email***	***password***
1	xFml4nujo4T ruaYkis3Zug==	luXFmrw0TJWZgC MGtvRP4g==	BtXVRODO8Whe ixRIYIyZTaFvbsi GtvdO9/b6U9+DJsc=	UH4/vo7Jk1 XJskRmq6KV/A==
2	emq+BTPSV5 6btja1qZBBLw==	GD+YoVEDVcJ ytmkfsc2uZQ==	g0GRjnVZ2Fc1R QGUeT3oC8clX/ CYGwsaVCdzxgwcJ4s=	HrbKDDFljI1HV oHTvyK8rQ==
3	auA338KreIW zg9vKZfVigw==	WiFVoCZKfOzk 4khz4M5J0A==	ril6bJXFhmgik ZWQOCm8TQ==	0fjyFVtGP+Uon NN4lpbXYw==
4	zMFSGvPh87qk 38CNz65RUA==	WbRzi7FAgXSC FjVyWGwbvg==	W9UII5rtTs3i pRXA+MITyA==	CHXi0Wg3DB6 3tVN501j2uQ==
5	eWBwf1wL9F2 WMbtayD7vZA==	Nk3DjbxyKlk1 VolFio0hLA==	UaYlb7NkWduyRH14KKTpTS+A2kl6 u7LNgG uAWVHso0Y=	UOXgw4/bbITpF KmWzQZ3ZA==

Dari ujicoba ini, didapatkan bahwa aplikasi VirCle berhasil melakukan proses enkripsi data pengguna dan menyimpannya ke basis data *Firebase*. Terlihat data pengguna yang tersimpan merupakan data yang telah terenkrip (*chipertext*). Selanjutnya, ujicoba juga memperlihatkan keberhasilan melakukan dekripsi data menjadi data asli kembali dan ditampilkan pada *homescreen*.

2. Evaluasi Waktu Proses

Evaluasi waktu proses dilakukan untuk mengetahui kecepatan proses enkripsi dan dekripsi data yang telah diimplementasikan. Meskipun studi terkait lainnya yang telah dilakukan memperlihatkan bahwa kecepatan proses enkripsi dan dekripsi data algoritma AES lebih cepat dibandingkan algoritma simestris lainnya seperti Twofish [19]. Hal ini dimaksudkan untuk mengantisipasi apakah penambahan proses enkripsi dan dekripsi data pengguna pada aplikasi VirCle mempengaruhi kinerja aplikasi secara keseluruhan. Pengukuran dilakukan dengan menyisipkan kode pencacah waktu, baik itu sebelum dan sesudah enkripsi serta proses dekripsi-nya. Hasil evaluasi menunjukkan waktu proses enkripsi dan dekripsi cukup cepat dengan rata-rata total waktu 1.35 milidetik untuk proses enkripsinya dan 1.53 mili detik untuk proses dekripsi. Waktu proses enkripsi dan dekripsi tertinggi yaitu 3 milidetik. Hasil pengukuran waktu proses untuk data uji 30 pengguna disajikan pada Gambar 5.



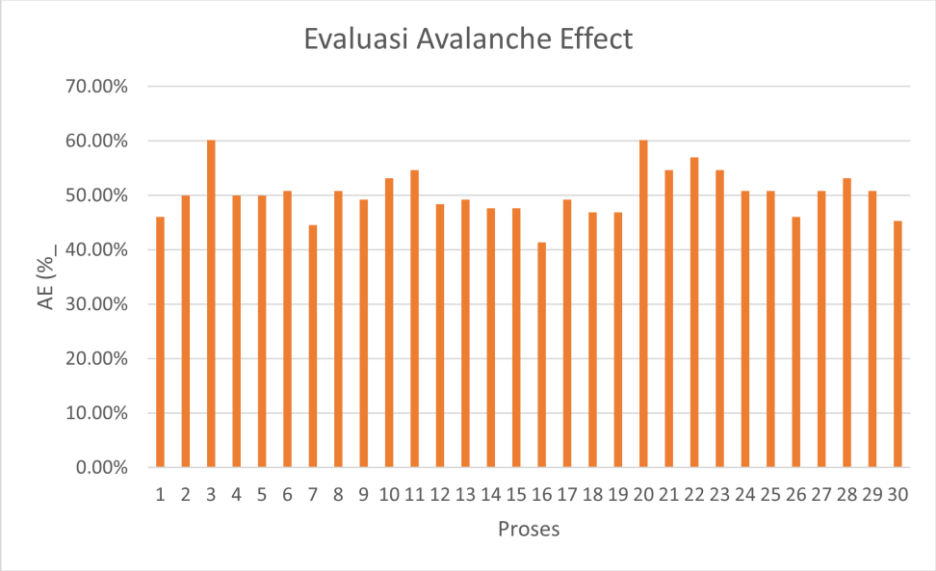
Gambar 3: Diagram Waktu Proses Enkripsi dan Dekripsi

3. Evaluasi *Avalanche Effect*

Avalanche Effect merupakan salah satu alternatif evaluasi yang dapat dilakukan untuk mengetahui tingkat keacakan hasil enkripsi. Proses evaluasi dilakukan dengan cara mengenkripsi salah satu data pengguna, setelah itu mengubah satu karakter dari data tersebut dan dienkripsi. Hasil dari kedua proses enkripsi akan dibandingkan lalu dihitung dengan rumus dari *Avalanche Effect* (AE) yang ditunjukkan pada (1).

$$AE = \frac{Jumlah\ bit\ yang\ berubah}{Total\ jumlah\ bit} \times 100\% \quad \ldots(1)$$

Berdasarkan evaluasi *AE*, algoritma yang baik memiliki nilai minimal 45% - 50%, dimana semakin besar persentasi yang didapat semakin baik tingkat keacakan yang dihasilkan sehingga meningkatkan kesulitan untuk memecahkan *chipertext* melalui analisis statistik atau *cryptanalysis* [20]–[23]. Dalam evaluasi ini, *field* data pengguna yaitu *fullname* dipilih sebagai data yang diuji. Hasil evaluasi diperoleh rata-rata persentasi sebesar 50.36% yang berarti baik. Gambar 17 menyajikan hasil evaluasi tingkat keacakan yang dilakukan.



Gambar 6: Diagram Hasil Evaluasi Avalanche Effect

IV. KESIMPULAN

Berdasarkan evaluasi yang dilakukan, memperlihatkan bahwa aplikasi VirCle yang mengimplementasikan algoritma AES-128 Bit dapat melakukan enkripsi dan dekripsi data pengguna yang tersimpan di basis data *Firebase*. Proses enkripsi dilakukan saat pengguna menyimpan data registrasi ke basis data, sementara proses dekripsi dikerjakan sebelum data ditampilkan pada *Home screen* aplikasi. Evaluasi kinerja aplikasi dari aspek kecepatan proses terhadap 30 (tiga puluh) data pengguna yang diuji, diketahui bahwa waktu proses enkripsi dan dekripsi data pengguna cukup cepat dengan rata-rata total waktu sebesar 1.35 milidetik untuk proses enkripsi dan 1.53 milidetik untuk proses dekripsinya. Selain itu, evaluasi *Avalanche Effect* yang dilakukan untuk mengukur tingkat keacakan dari hasil enkripsi algortima AES-128 bit yang digunakan, didapat rata-rata persentasi sebesar 50.36%, yang berarti tingkat keacakan data dipandang baik. Adapun untuk penelitian selanjutnya yaitu dapat dikaji pemakaian proses atau teknik pembangkitan kunci lainnya untuk memperoleh kunci yang lebih unik dengan tujuan menaikkan tingkat keacakan dari hasil enkripsi algoritma AES.

DAFTAR PUSTAKA

[1] A. S. Cahyono, "Pengaruh Media Sosial Terhadap Perubahan Sosial Masyarakat Di Indonesia," *publiciana*, vol. 9, no. 1, pp. 140–157, 2016.
[2] T. Liedfray, F. J. Waani, and J. J. Lasut, "Peran Media Sosial Dalam Mempererat Interaksi Antar Keluarga Di Desa Esandom Kecamatan Tombatu Timur Kabupaten Minahasa Tenggara," *Jurnal Ilmiah Social*, vol. 2, no. 1, 2022.
[3] R. C. Halim and S. Sugiarto, "Penerapan Algoritma AES dalam Perancangan Aplikasi Media Sosial Berbasis Android," *ENTER*, vol. 1, 2018.
[4] L. Mustika, "Implementasi Algoritma AES Untuk Pengamanan Login Dan Data Customer Pada E-Commerce Berbasis Web," *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 1, p. 148, Feb. 2020, doi:

10.30865/jurikom.v7i1.1943.
[5] M. H. Rumlus and H. Hartadi, "Kebijakan Penanggulangan Pencurian Data Pribadi dalam Media Elektronik," *Jurnal HAM*, vol. 11, no. 2, p. 285, 2020, doi: 10.30641/ham.2020.11.285-299.
[6] R. Mulud Muchamad, A. Asriyanik, and A. Pambudi, "Implementasi Algoritma Advanced Encryption Standard (Aes) Untuk Mengenkripsi Datastore Pada Aplikasi Berbasis Android," *Jurnal Mnemonic*, vol. 6, no. 1, pp. 55–64, 2023, doi: 10.36040/mnemonic.v6i1.5889.
[7] N. Matondang, I. N. Isnainiyah, and A. Muliawatic, "Analisis Manajemen Risiko Keamanan Data Sistem Informasi (Studi Kasus: RSUD XYZ)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 2, no. 1, pp. 282–287, 2018, doi: 10.29207/resti.v2i1.96.
[8] D. Hulu, B. Nadeak, and S. Aripin, "Implementasi Algoritma AES (Advanced Encryption Standard) Untuk Keamanan File Hasil Radiologi di RSU Imelda Medan," *KOMIK (Konferensi …*, vol. 4, pp. 78–86, 2020, doi: 10.30865/komik.v4i1.2590.
[9] D. A. Meko, "Perbandingan Algoritma DES, AES, IDEA Dan Blowfish dalam Enkripsi dan Dekripsi Data," *Jurnal Teknologi Terpadu*, vol. 4, no. 1, 2018, doi: 10.54914/jtt.v4i1.110.
[10] A. P. Sidik *et al.*, "Teknik Xor Pada Mode Operasi Algoritma Cipher Block Chaining (Cbc) Dengan Kunci Acak Blum Blum Shub Dalam Meningkatkan Keamanan Data," *Jurnal Mantik Penusa*, vol. 3, no. 2, pp. 130–135, 2019.
[11] D. Andriani, "Perancangan Aplikasi Penyandian Teks Dengan Menggunakan Algoritma Chiper Block Chaining," *Jurnal Teknik Informatika Unika St. Thomas (JTIUST)*, vol. 02, no. 338, pp. 14–23, 2017, [Online]. Available: http://www.ejournal.ust.ac.id/index.php/JTIUST/article/view/186%0Ahttp://www.ejournal.ust.ac.id/index.php/JTIUST/article/download/186/189
[12] A. Teguh Utomo and R. Pradana, "Implementasi Algoritma Advanced Encryption Standard (AES-128) Untuk Enkripsi dan Dekripsi File," *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia*, no. September, pp. 21–23, 2022.
[13] R. K. Endrayanto, A. Muttaqin, and R. A. Setyawan, "Advanced Encryption Standard (AES) pada Modul Internet of Things (IoT)," *TELKA - Telekomunikasi, Elektronika, Komputasi dan Kontrol*, vol. 5, no. 2, pp. 103–113, 2019, doi: 10.15575/telka.v5n2.103-113.
[14] F. Muharram, H. Azis, and A. R. Manga, "Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES)," *Proc. of the Seminar Nasional Ilmu Komputer dan Teknologi Informasi*, vol. 3, no. 2, pp. 112–115, 2018.
[15] V. Yuniati, G. Indriyanta, and A. Rachmat C., "Enkripsi Dan Dekripsi Dengan Algoritma Aes 256 Untuk Semua Jenis File," *Jurnal Informatika*, vol. 5, no. 1, 2011, doi: 10.21460/inf.2009.51.69.
[16] A. Prameshwari and N. P. Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," *Eksplora Informatika*, vol. 8, no. 1, p. 52, 2018, doi: 10.30864/eksplora.v8i1.139.
[17] B. Dan and T. Pada, "Analisis Perbandingan Unjuk Kerja Algoritma Comparative Analysis of Des , 3Des , Aes , Blowfish and Twofish Algorithm Performance on Documents," *Repository.Usd.Ac.Id*, 2019, [Online]. Available: https://repository.usd.ac.id/36632/2/155314086_full.pdf
[18] D.-J. Munoz, J. A. Montenegro, M. Pinto, and L. Fuentes, "Energy-aware environments for the development of green applications for cyber–physical systems," *Future Generation Computer Systems*, vol. 91, pp. 536–554, 2019, doi: https://doi.org/10.1016/j.future.2018.09.006.
[19] E. E. A. E. H. E. H. N. T. N. T. N. Padilah, "Analisis Perbandingan Hasil Enkripsi Dan Deskripsi Algoritma Kriptografi Rijndael Dan Twofish Untuk Penyandian Data," *Jurnal Mahasiswa Ilmu Komputer*, no. Vol 3 No 1 (2022): JMIK Maret 2022, pp. 260–265, 2022, [Online]. Available: https://scholar.ummetro.ac.id/index.php/IlmuKomputer/article/view/1918/932
[20] I. Fitriani and A. B. Utomo, "Implementasi Algoritma Advanced Encryption Standard (AES) pada Layanan SMS Desa," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 5, no. 3, pp. 153–163, 2020, doi: 10.14421/jiska.2020.53-03.
[21] R. R. Fauzi and T. Wellem, "Perancangan Kriptografi Block Cipher berbasis Pola Dribbling Practice," *Aiti*, vol. 18, no. 2, pp. 158–172, 2021, doi: 10.24246/aiti.v18i2.158-172.
[22] F. Rizky, "Implementasi Kriptografi Dengan Metode Advanced Encryption Standard (AES) Untuk Realtime Chat Berbasis Mobile Pada E - Learning Politeknik Negeri Lhokseumawe," *JAISE (Journal of Artificial Intelligence and Software Engineering)*, vol. 1, no. 2, pp. 1–8, 2019.

[23] A. P. Putra, H. Herfina, S. Maryana, and A. Setiawan, “Implementasi Algoritma AES (Advance Encryption Standard) Rijndael Pada Aplikasi Keamanan Data,” *JIPETIK: Jurnal Ilmiah Penelitian Teknologi Informasi & Komputer*, vol. 1, no. 2, pp. 46–51, 2020.