SKRIPSI

IMPLEMENTASI ALGORITMA ELLIPTIC CURVE CRYPTOGRAPHY (ECC) DENGAN END-TO-END ENCRYPTION PADA APLIKASI CHAT BERBASIS MOBILE

Disusun dan diajukan oleh:

MUHAMMAD RIDHOI
D121 18 1303

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2023

LEMBAR PENGESAHAN SKRIPSI

IMPLEMENTASI ALGORITMA ELLIPTIC CURVE CRYPTOGRAPHY (ECC) DENGAN END-TO-END ENCRYPTION PADA APLIKASI CHAT BERBASIS MOBILE

Disusun dan diajukan oleh

MUHAMMAD RIDHOI
D121181303

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas HasanuddinPada tanggal 22 Februari 2023 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.
NIP. 19750313 200912 1 003

Pembimbing Pendamping,

Dr. Eng. Muhammad Niswar, S.T., M.IT.
NIP. 19730922 199903 1 001

Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, ST, MT,M.Bus.Sys.,IPM, ASEAN
NIP 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama	: Muhammad Ridhoi
NIM	: D121181303
Departemen	: Teknik Informatika
Jenjang	: S1

Menyatakan dengan ini karya tulisan saya berjudul:

"IMPLEMENTASI ALGORITMA *ELLIPTIC CURVE CRYPTOGRAPHY* (ECC) DENGAN *END-TO-END ENCRYPTION* PADA APLIKASI *CHAT* BERBASIS *MOBILE*"

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilalihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 23 Februari 2023

Yang menyatakan,

Muhammad Ridhoi

ABSTRAK

MUHAMMAD RIDHOI. *Implementasi Algoritma Elliptic Curve Cryptography (ECC) Dengan End-To-End Encryption Pada Aplikasi Chat Berbasis Mobile* (dibimbing oleh Dr. Eng. Ady Wahyudi Paundu, S.T., M.T. dan Dr. Eng. Muhammad Niswar, S.T., M.IT.)

Perkembangan aplikasi *mobile* kini sudah tak terkedali. Dapat dilihat dari banyaknya aplikasi yang mampu membantu memenuhi kebutuhan manusia, salah satunya dalam melakukan komunikasi yaitu menggunakan aplikasi *Chatting*. Namun pesan yang dikirim melalui aplikasi *Chatting* sering berisi informasi pesan yang penting bahkan rahasia dan harus dijaga keamanannya dari penyalahgunaan oleh pihak yang tidak berwenang. Salah satu cara yang bisa digunakan untuk menjaga keamanan data ialah kriptografi, dimana terdapat suatu proses data yang dikirim akan disandikan dengan proses enkripsi dan dekripsi. Metode kriptografi yang cocok pada perangkat *mobile* salah satunya adalah *Elliptic Curve Cryptography* (ECC) yang dapat mengurangi biaya komputasi karena metode perkaliannya yang tercepat sehingga menghasilkan pengurangan *overhead* pada proses komputasi. Pada penelitian ini bertujuan untuk membangun aplikasi *chat* berbasis *mobile* yang mengimplementasikan algoritma kriptografi *Elliptic Curve Cryptography* (ECC) untuk mengamankan pesan. Hasil implementasi enkripsi dan dekripsi algoritma *Elliptic Curve Cryptography* (ECC) pada aplikasi *chat* dengan pemantauan *request* yang dikirim ke server, didapati bahwa pesan yang kirim sudah dalam bentuk *ciphertext* sehingga tidak mudah dibaca. Hasil uji kecepatan waktu proses enkripsi dan dekripsi algoritma ECC pada parameter kurva eliptik yang berbeda menunjukkan parameter Secp192r1 28.9% dan 77.1% lebih cepat dibandingkan dengan Secp256r1 dan Secp521r1 pada proses enkripsi, sedangkan pada proses dekripsi 27,9% dan 73.5% lebih cepat dibandingkan Secp256r1 dan Secp521r1. Perbedaan waktu proses enkripsi dan dekripsi dari setiap parameter yang berbeda disebabkan oleh besarnya *overhead* yang ditentukan oleh panjang karakter pesan dan nilai parameter yang digunakan dalam algoritma *Elliptic Curve Cryptography* (ECC).

Kata kunci: Aplikasi *Chat*, Kriptografi, Enkripsi, *Elliptic Curve Cryptography* (ECC)

ABSTRACT

MUHAMMAD RIDHOI. *Implementation of Elliptic Curve Cryptography (ECC) Algorithm with End-To-End Encryption in Mobile-Based Chat Application* (supervised by Dr. Eng. Ady Wahyudi Paundu, S.T., M.T. and Dr. Eng. Muhammad Niswar, S.T., M.IT.)

The development of mobile applications has become unstoppable, as evidenced by the increasing number of applications that help meet human needs, such as communication using Chatting applications. However, messages sent through Chatting applications often contain important or even confidential information that must be kept secure from unauthorized parties. Cryptography is one way to maintain data security, where data sent is encrypted and decrypted. One suitable cryptography method for mobile devices is Elliptic Curve Cryptography (ECC), which can reduce computational costs due to its fast multiplication method, resulting in a reduction in overhead in the computation process. This research aims to develop a mobile-based chat application that implements the Elliptic Curve Cryptography (ECC) algorithm to secure messages. The implementation results of the encryption and decryption algorithms of ECC on the chat application with monitoring of requests sent to the server showed that the sent messages were already in ciphertext form, making it difficult to read. The results of the speed test for the encryption and decryption processes of the ECC algorithm using different elliptic curve parameters showed that the Secp192r1 parameter was 28.9% and 77.1% faster than the Secp256r1 and Secp521r1 parameters for the encryption process, while for the decryption process, it was 27.9% and 73.5% faster than Secp256r1 and Secp521r1. The difference in the encryption and decryption process time for each different parameter is due to the overhead size determined by the length of the message characters and the parameter values used in the Elliptic Curve Cryptography (ECC) algorithm.

Keywords: Chat Application, Cryptography, Encryption, Elliptic Curve Cryptography (ECC)

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
a, b	Bilangan konstanta
m	Gradien
mod	Operasi modulus
p	Bilangan prima
x, y	Koordinat titik
P, Q	Titik pada kurva eliptik
k	Skalar, bilangan bulat positif
G	Titik dasar pada kurva eliptik

DAFTAR LAMPIRAN

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat dan hidayah yang telah diberikan, sehingga tugas akhir ini bisa diselesaikan. Penulis menyadari bahwa penyelesaian tugas akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan terima kasih banyak kepada pihak-pihak yang telah banyak memberikan bantuan, dorongan baik moral maupun spiritual. Ucapan terima kasih penulis tujukan kepada:

1. Kedua orang tua penulis, Bapak dan Ibu beserta keluarga atas segala doa, dukungan, semangat, pengorbanan, dan kasih sayang yang telah diberikan.
2. Bapak Dr. Eng. Ady Wahyudi Paundu, ST., M.T. selaku Dosen Pembimbing I yang telah meluangkan waktu dan pikiran dalam memberikan bimbingan dan masukan yang sangat bermanfaat dalam penyusunan laporan skripsi ini.
3. Bapak Dr. Eng. Muhammad Niswar, S.T., M.IT. selaku dosen Pembimbing II yang telah meluangkan waktu dan pikiran dalam memberikan bimbingan dan masukan yang sangat bermanfaat dalam penyusunan laporan skripsi ini.
4. Seluruh Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.
5. Para sahabat penulis yang telah memberikan motivasi, nasihat, dukungan, dan semangat selama proses perkuliahan.
6. Teman-teman SYNCHRONOUS selaku rekan belajar sejak awal hingga akhir masa perkuliahan.
7. Seluruh pihak terkait yang tidak dapat penulis sebutkan satu persatu yang telah membantu penulis menyelesaikan penulisan skripsi ini.

Semoga Allah SWT membalas seluruh kebaikan yang telah mereka berikan kepada penulis. Penulis juga menyadari bahwa masih terdapat banyak kekurangan pada penulisan tugas akhir ini. Oleh karena itu, penulis meminta maaf atas segala

kekurangan yang ada pada karya tulis ini. Kritik dan saran sangat penulis harapkan bagi penyempurnaan tugas akhir ini. Semoga tugas akhir ini bisa memberikan manfaat bagi penulis dan pihak lain yang membacanya.

Makassar, Februari 2023

Penulis

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan aplikasi *mobile* kini sudah tak terkedali. Dapat dilihat dari banyaknya aplikasi yang mampu membantu memenuhi kebutuhan dan kepentingan manusia, salah satunya dalam melakukan komunikasi yaitu menggunakan aplikasi *Chatting*. Pengiriman pesan melalui internet dengan menggunakan aplikasi *Chatting* merupakan salah satu metode komunikasi yang bersifat *real-time*. Namun pesan yang dikirim melalui aplikasi *Chatting* sering berisi informasi pesan yang penting bahkan rahasia dan harus dijaga keamanannya dari penyalahgunaan oleh pihak yang tidak berwenang. Aplikasi *Chatting* juga memerlukan sistem keamanan yang baik untuk mengamankan pesan instan bersifat rahasia dan eksklusif. Masalah penyalahgunaan pesan instan pada aplikasi yang berjalan di jaringan internet juga menjadi isu krusial yang harus ditemukan solusinya (Randi dkk., 2020).

Pentingnya menjaga kerahasiaan data yang ada agar mencegah penyalahgunaan pesan yang dikirimkan merupakan salah satu hal yang perlu diperhatikan. Salah satu cara yang bisa digunakan untuk menjaga keamanan data ialah kriptografi. Menurut (Fatonah dkk., 2022) di dalam kriptografi terdapat suatu proses di mana data atau informasi yang dikirimkan akan disandikan (enkripsi dan dekripsi). Enkripsi ini dilakukan saat informasi akan dikirimkan dengan penyandian terlebih dahulu sehingga informasi tersebut akan sulit terbaca. Kemudian proses Dekripsi dilakukan saat penerimaan informasi dengan cara mengubah kembali menjadi bentuk aslinya. Proses Dekripsi hanya bisa dilakukan penerima dengan menggunakan kunci rahasia yang telah disepakati bersama sebelumnya. Kriptografi salah satu cara yang efektif untuk mengatasi ancaman-ancaman terhadap keamanan informasi data, sehingga meskipun data dicuri, informasinya tidak dapat dibaca karena telah disandikan atau dikodekan dengan metode tertentu (Panggabean, 2020). Salah satu metode kriptografi yang dapat mengakomodasi hal tersebut ialah *Elliptic Curve Cryptography* (ECC).

Elliptic Curve Cryptography (ECC) adalah salah satu pendekatan algoritma kriptografi kunci publik yang menggunakan kurva eliptik dengan semua variabel dan koefisiennya terbatas pada elemen dari suatu Galois Field. *Elliptic Curve Cryptography* (ECC) juga merupakan teknik kriptografi asimetri yang menggunakan dua buah kunci berbeda dalam proses enkripsi dan dekripsi. Kedua kunci tersebut dikenal dengan *private key* yang digunakan untuk dekripsi data dan *public key* yang digunakan untuk enkripsi data (Damanik, 2019). *Elliptic Curve Cryptography* (ECC) menjadi salah satu pendekatan untuk kemanan siber yang sangat menarik pada perangkat kecil seperti *smartphone*, meteran dan *embedded devices* (Diro dkk., 2017) dengan mekanisme keamanannya dapat mengurangi biaya komputasi, data yang dikirim dan disimpan, karena metode perkaliannya yang tercepat dan memiliki panjang kunci yang pendek (Qazi dkk., 2021).

Elliptic Curve Cryptography (ECC) memiliki kelebihan yaitu memiliki keamanan yang sama dengan algoritma yang lain pada distribusi kunci akan tetapi dengan manajemen kunci lebih baik karena ukuran kunci yang lebih kecil jika dibandingkan dengan algoritma yang lain dan usia dari algoritma ECC ini lebih baru (Pratiwi & Asmunin, 2022). Misalnya, Elliptic Curve Cryptography (ECC) kunci 160 bits setara dengan Rivest Shamir Adleman (RSA) kunci 1024 bits, ECC kunci 224 bits setara dengan RSA kunci 2048 bits. Sehingga *Elliptic Curve Cryptography* (ECC) dapat diterapkan secara luas pada perangkat dengan ruang penyimpanan dan *bandwidth* yang terbatas, karena panjang kunci yang pendek, kecepatan tinggi, dan konsumsi daya yang rendah (Perdana dkk., 2022). Dengan demikian menghasilkan pengurangan *overhead* pada proses komputasi menjadi faktor penting yang membuat ECC menjadi pilihan yang lebih baik dari pada RSA, DSA dan AES (Qazi dkk., 2021).

1.2 Rumusan Masalah

Bagaimana menjaga kerahasiaan dan keamanan pesan teks dari penyalahgunaan oleh pihak yang tidak berwenang dengan *Elliptic Curve Cryptography* (ECC) secara *End-to-End* pada aplikasi *Chat* berbasis *Mobile*?

1.3 Tujuan Penelitian

1. Mengimplementasikan sistem kriptografi algoritma *Elliptic Curve Cryptography* (ECC) pada aplikasi *Chat* berbasis *Mobile* dalam menjaga kerahasiaan dan keamanan pesan teks secara *End-to-End* dari penyalahgunaan oleh pihak yang tidak berwenang.
2. Menganalisa parameter standar kriptografi kurva eliptik Secp192r1, Secp256r1, dan Secp521r1 pada enkripsi dan dekripsi algoritma *Elliptic Curve Cryptography* (ECC).

1.4 Manfaat Penelitian

1. Membantu masyarakat dalam menjaga kerahasiaan dan keamanan pesan teks yang dikirim maupun diterima tetap aman.
2. Mengetahui sejauh mana peran *Elliptic Curve Cryptography (ECC)* dalam menjaga kerahasiaan dan keamanan pesan teks.
3. Menjadi dasar untuk penelitian terkait sistem kripto dengan *Elliptic Curve Cryptography (ECC)*.

1.5 Ruang Lingkup

1. Pengembangan aplikasi *Chat* menggunakan *framework* Flutter berbasis Android.
2. Menggunakan bahasa pemrograman Dart.
3. Karakter pada pesan teks yang di enkripsi ialah ASCII.
4. Informasi yang dienkripsi dan didekripsi berupa pesan teks.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini akan dijelaskan teori-teori yang menunjang percobaan yang dilakukan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi analisis kebutuhan sistem, perancangan implementasi sistem, implementasi algoritma, skenario penggunaan dan skenario pengujian.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Bab ini berisi hasil penelitian dan pembahasan.

BAB V PENUTUP

Bab ini berisi kesimpulan hasil penelitian dan saran

BAB II
TINJAUAN PUSTAKA

2.1 Kriptografi

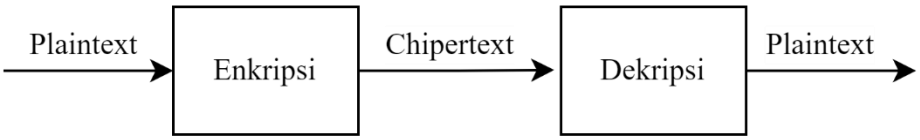
Kriptografi berasal dari gabungan dua kata yaitu “Crypto” yang berarti rahasia dan “graphy” yang berarti tulisan. Dalam bahasa komputasi kriptografi diartikan sebagai ilmu dan seni untuk menjaga keamanan data. Ahli kriptografi disebut kriptografer. Kriptografi merupakan salah satu cara untuk mencegah kebocoran data yang bersifat rahasia, dimana dalam memproteksi pengiriman data dengan mengubahnya menjadi kode tertentu dan hanya ditujukan untuk orang yang hanya memiliki sebuah kunci untuk mengubah kode itu kembali yang berfungsi dalam menjaga kerahasiaan data atau pesan. Dalam kriptografi, data atau pesan yang dikirimkan melalui jaringan akan disamarkan sedemikian rupa. Sehingga seandainya data tersebut bisa diperoleh dan dibaca oleh orang lain, maka pihak yang tidak berhak atau berwenang tersebut tidak akan bisa mengerti arti dari data tersebut (Amrulloh & Ujianto, 2019).

Beberapa istilah yang digunakan dalam bidang kriptografi yaitu:

1. *Plaintext* adalah pesan asli yang hendak dikirimkan (berisi data asli) berupa kumpulan karakter yang dapat berupa abjad, angka atau simbol tertentu yang dapat dibaca.
2. *Chiphertext* adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi dari algoritma enkripsi yang tidak dapat dibaca secara langsung.
3. Enkripsi, salah satu proses utama dalam kriptografi dimana suatu proses di mana sebuah pesan asli (*plaintext*) diubah menjadi bentuk pesan lain yang tidak dapat dibaca (*ciphertext*) menggunakan suatu fungsi matematis kunci tertentu yang disebut key, dalam upaya pengamanan data yang dikirimkan terjaga rahasianya.
4. Dekripsi, merupakan kebalikan dari enkripsi, dimana ciphertext dirubah kembali ke plaintext dengan menggunakan fungsi matematis dan key, atau proses pesan yang telah dienkripsi dikembalikan seperti semula.

5. Kunci, atau *key* digunakan dalam proses melakukan enkripsi dan dekripsi. *Key* terbagi menjadi dua bagian yaitu kunci privat dan kunci umum atau kunci publik.

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Enkripsi adalah suatu proses yang melakukan perubahan dari suatu pesan yang bisa dibaca menjadi tidak dapat dibaca. Dekripsi adalah suatu proses untuk mengembalikan informasi yang tidak bisa dimengerti tadi menjadi seperti semula (Amrulloh & Ujianto, 2019). Proses enkripsi dan dekripsi kriptografi dapat dilihat pada gambar 1.



Gambar 1 Proses Ekripsi dan Dekripsi

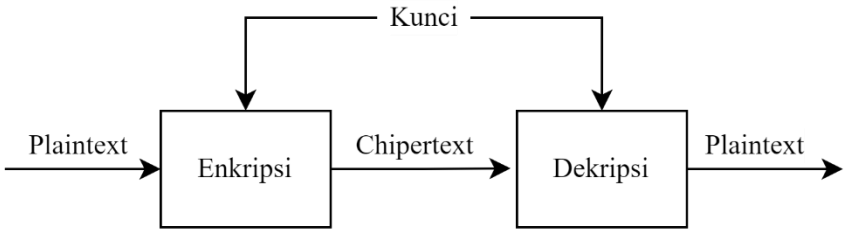
Secara umum ada dua jenis kriptografi, yaitu kriptografi klasik dan kriptografi modern, Kriptografi klasik (simetris) adalah suatu algoritma yang menggunakan satu kunci untuk mengamankan data. Dua teknik dasar yang biasa dilakukan adalah substitusi dan transposisi. Sedangkan kriptografi modern (asimetrik) adalah algoritma yang lebih kompleks dari pada algoritma klasik, hal ini disebabkan algoritma ini menggunakan komputer (Sumandri, 2017).

2.1.1 Kriptografi Simetris

Kriptografi simetri atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi (Sumandri, 2017). Metode menyembunyikan pesannya adalah dengan teknik substitusi atau transposisi atau keduanya. Dalam kriptografi simetri, kunci enkripsi yang digunakan sama dengan kunci dekripsi sehingga algoritma ini disebut juga sebagai *single-key algorithm*. Keamanan dari pesan yang menggunakan kriptografi simetri ini tergantung pada kuncinya, jika kunci tersebut diketahui oleh orang lain maka orang tersebut akan dapat melakukan enkripsi dan dekripsi terhadap pesan tersebut. Jadi pembuat

pesan dan penerimanya harus memiliki kunci yang sama persis. Siapapun yang memiliki kunci tersebut, termasuk pihak-pihak yang tidak diinginkan dapat membuat dan membongkar rahasia *ciphertext* (Basri, 2016)

Secara sederhana proses enkripsi dan dekripsi dengan kriptografi simetri dapat digambarkan pada gambar 2.



Gambar 2 Skema Kriptografi Simetri

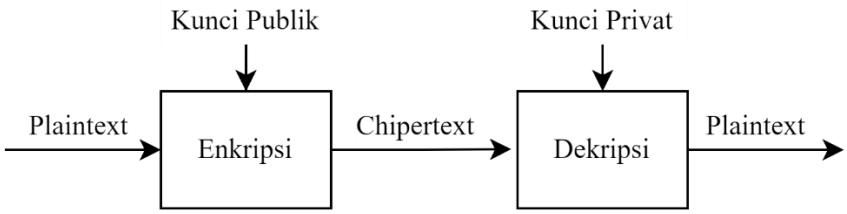
2.1.2 Kriptografi Asimetris

Kriptografi Asimetris atau biasa juga disebut dengan algoritma kriptografi kunci publik yaitu kriptografi yang menggunakan dua buah kunci yang berbeda dalam proses enkripsi dan deskripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsikan suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia tertentu dalam hal ini kunci private untuk melakukan pembongkaran terhadap sandi yang dikirim untuknya (Basri, 2016).

Gambaran umum kriptografi asimetris dapat dilihat pada gambar 3.

Pada kriptografi asimetris, kuncinya terbagi menjadi dua bagian yaitu:

1. Kunci publik (*public key*) yaitu kunci yang dapat mengenkripsi data dan boleh didistribusikan secara luas tanpa memengaruhi keamanan.
2. Kunci privat (*private key*) yaitu kunci yang dirahasiakan atau hanya pemakai yang mempunyai akses kunci dapat mengdekripsi data enkripsi tersebut.

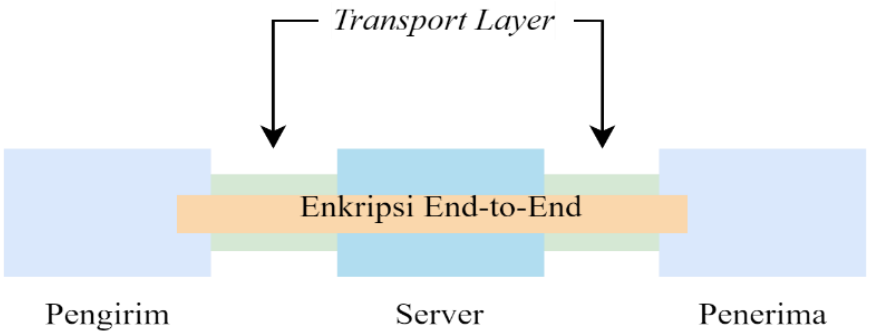


Gambar 3 Skema Kriptografi Asimetris

2.1.3 Sistem Enkripsi *End-to-End*

Enkripsi *end-to-end (End-to-end Encryption)* adalah salah satu yang paling banyak digunakan untuk pengamanan pengiriman informasi melalui internet. Pada dasarnya, enkripsi *end-to-end* merupakan jenis enkripsi dalam suatu sistem komunikasi yang dilakukan terhadap pesan sebelum dikirim oleh pengirim dan kembali didekripsi saat pesan sampai ke penerima, dalam artian enkripsi data dilakukan pada sumber pengiriman pesan dan kemudian didekripsi hanya pada tujuan akhir atau penerima pesan. Dengan enkripsi *end-to-end*, informasi dapat dikirim melalui jaringan dengan satu jalur yang hanya pengirim dan penerima yang bisa mengaksesnya sehingga informasi yang dikirim dari pengirim pesan dikemas dalam bentuk kunci khusus yang hanya bisa dideskripsikan oleh penerima (Santria & Arsoetar, 2017).

Adapun ilustrasi dari enkripsi *end-to-end* dapat dilihat pada gambar 4.



Gambar 4 Ilustrasi Enkripsi *End-to-End*

2.2 Algoritma *Elliptic Curve Cryptography*

Elliptic Curve Cryptography atau Kriptografi Kurva Eliptik adalah sebuah algoritma kriptografi kunci publik, yaitu algoritma dimana setiap pihaknya memiliki sepasang kunci privat dan kunci publik. Kunci privat hanya dimiliki oleh pribadi-pribadi yang berkepentingan, sedangkan kunci publik disebarluaskan ke semua pihak (Santoso & Siambaton, 2020). *Elliptic Curve Cryptography* menjadi salah satu pendekatan kriptografi kunci asimetris yang mendasarkan keamanannya pada persoalan logaritma diskrit dari kurva eliptik bidang terbatas. Struktur kurva eliptik digunakan sebagai grup operasi matematis untuk melangsungkan proses enkripsi dan dekripsi. Salah satu kegunaan ECC yaitu skema enkripsi yang menggunakan algoritma ECC ElGamal (Nugroho & Munir, 2015).

Kurva ellips dalam kriptografi dicetuskan oleh Neal Koblitz dan Victor S. Miller pada tahun 1985. Kurva ellips juga digunakan pada beberapa algoritma pemfaktoran integer yang juga memiliki aplikasinya dalam kriptografi, seperti *Lenstra Elliptic Curve Factorization*. Algoritma kunci publik ini, berdasarkan pada variasi perhitungan matematis yang terbilang sangat sulit dipecahkan tanpa pengetahuan tertentu mengenai bagaimana perhitungan tersebut dibuat (Mardianto dkk., 2015). Pendekatan yang dilakukan untuk menghasilkan algoritma Kriptografi Kurva Eliptik adalah dengan menggunakan struktur matematika yang sangat unik yang memungkinkan pemrosesan titik dengan memiliki dua buah titik dalam sebuah kurva eliptik dan menghasilkan sebuah titik lain yang ada pada kurva tersebut. Struktur yang unik ini memberikan keuntungan dalam kriptografi dikarenakan kesulitan untuk menemukan 2 buah titik yang menentukan sebuah titik tertentu tersebut tidak dapat ditemukan dengan mudah. Tingkat kesulitan untuk menemukan 2 buah titik termasuk dalam golongan yang rumit sama seperti kesulitan untuk memperhitungkan 4 variasi eksponensial yang digunakan dalam algoritma RSA yang telah banyak diimplementasikan. Untuk memecahkan Kriptografi Kurva Eliptik sendiri dibutuhkan perhitungan matematis yang sangat tinggi (Santoso & Siambaton, 2020).

Elliptic Curve Cryptography (ECC) mempunyai keuntungan jika dibandingkan dengan kriptografi asimetris lainnya yaitu dalam hal ukuran panjang kunci yang lebih pendek tetapi memiliki tingkat keamanan yang sama. Sebagai perbandingan, 160 bit *Elliptic Curve Cryptography* mempunyai tingkat keamanan (3.8.1010 MIPS/*Million Instruction per Second year*) yang sama dengan 1024 bit RSA mempunyai tingkat keamanan (3.1012 MIPS year). Sehingga kecepatannya lebih tinggi, konsumsi daya yang lebih rendah, adanya penghematan *bandwith* (Br Sembiring, 2015).

Kumpulan titik pada kurva dapat membentuk kumpulan abelian (dengan titik pada tak terhingga sebagai elemen identitas). Jika nilai x dan y dipilih dari daerah finite yang besar, solusi akan membentuk suatu kumpulan abelian finite. Permasalahan logaritma diskrit pada kumpulan kurva ellips tersebut dipercaya lebih sulit dibandingkan permasalahan yang sama (perkalian bilangan tidak nol) dalam daerah finite. Sebagai salah satu kriptosistem kunci publik, belum ada pembuktian matematis untuk tingkat kesulitan dari ECC yang telah dipublikasikan sampai tahun 2006. Tetapi U.S. National Security Agency telah mengesahkan teknologi ECC sebagai algoritma kriptografi yang dianjurkan (Mardianto dkk., 2015)

Medan berhingga atau biasa disebut juga sebagai *Galois Field* (GF) yaitu medan himpunan bilangan yang memiliki bilangan yang terbatas. *Order* dari medan berhingga yaitu jumlah banyaknya elemen yang ada dalam medan tersebut. Medan berhingga dilambangkan sebagai GF(q), dimana q adalah derajat medan berhingga yang berupa pangkat prima. Dalam merepresentasikan elemen Fq dengan $q = p^m$ maka p disebut sebagai karakteristik dari Fq dan m disebut sebagai derajat perluasan dari Fq, dimana p adalah bilangan prima dan m adalah bilangan bulat positif. Medan berhingga yang digunakan dalam *Elliptic Curve Cryptography* adalah Fq dengan $q = p$ dan $q = 2^m$ dengan $q = p$ disebut bidang prima dan dinotasikan sebagai Fp.

2.2.1 Operasi Matematika *Elliptic Curve Cryptography*

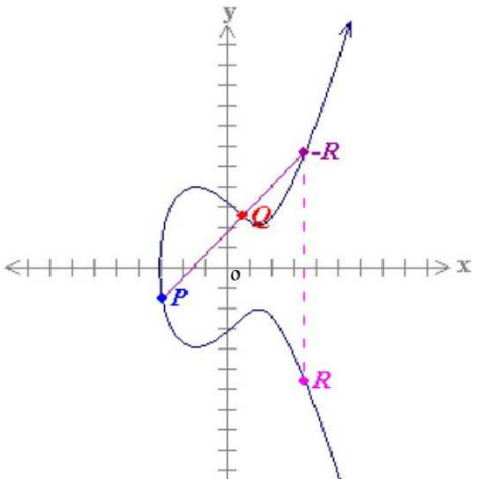
Persamaan matematika dari kurva eliptik pada bidang prima Fp yang digunakan pada *Elliptic Curve Cryptography* adalah sebagai berikut:

$x^3 + ax + b \; mod \; p$, dimana $4a^3 + 27b^2 \; mod \; p \; \neq \; 0$ (1)

Pada elemen bidang berhingga menggunakan bilangan bulat antara 0 dan p – 1. Aritmatika modular pada semua operasi seperti penambahan, pengurangan, pembagian, dan perkalian melibatkan bilangan bulat antara 0 dan p – 1. Bilangan prima p ditentukan dengan nilai tertentu, sehingga ada sebagian besar titik pada kurva eliptik memberikan sistem kriptografi yang aman. *Elliptic Curve Cryptography* memiliki tiga operasi utama yaitu penjumlahan titik, penggandaan titik, dan perkalian titik (Boruah & Saikia, 2014).

2.2.1.1 Penjumlahan Titik

Penjumlahan titik adalah penambahan dua titik berbeda yang menghasilkan koordinat titik ketiga. Koordiant titik ketika dihasilkan dari titik pada kurva eliptik yang dilalui garis yang ditarik dari koordinat dua titik tersebut dan diinvers terhadap sumbu x, sehingga sumbu y negatif dari y titik tersebut. Penjumlahan dua titik pada kurva eliptik dapat dilihat pada gambar 5.



Gambar 5 Penjumlahan dua titik pada kurva eliptik

Dua titik $P(x_1, y_1)$ dan $Q(x_2, y_2)$ adalah titik terpisah. $P + Q = R(x_3, y_3)$, dicari dengan perhitungan:

$$x_3 = m^2 - x_1 - x_2 \; mod \; p \qquad (2)$$

$$y_3 = m(x_1 - x_3) - y_1 \; mod \; p \qquad (3)$$

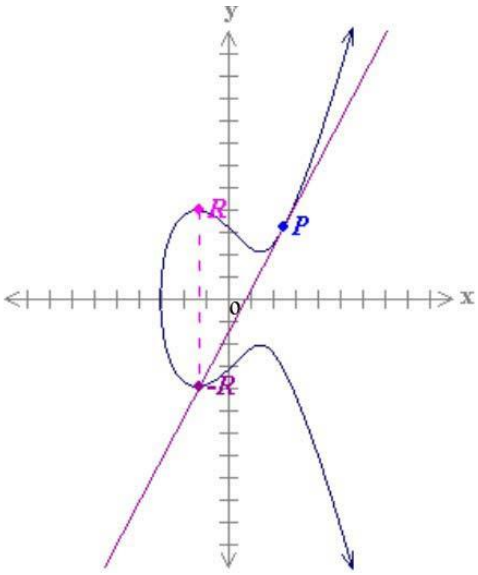
Dimana gradien (m):

$$m = \frac{y_2 - y_1}{x_2 - x_1} \qquad (4)$$

Dimana (x_1, y_1) koordinat titik pertama, (x_2, y_2) titik kedua dan (x_3, y_3) titik ketiga hasil perhitungan (Boruah & Saikia, 2014).

2.2.1.2 Penggandaan Titik

Penggandaan titik merupakan penjumlahan titik yang dijumlahkan dengan dirinya sendiri atau pada koordinat yang sama. Penggandaan titik membentuk tangen pada titik $P(x_1, y_1)$. Penggandaan titik pada kurva eliptik dapat dilihat pada gambar 6.



Gambar 6 Pengandaan Titik pada Kurva Eliptik

Penggandaan titik $P(x_1, y_1)$ dalam kurva eliptik, garis singgung ke kurva harus melintasi kurva melalui titik lain, ditulis sebagai $-R$ kemudian merefleksikan titik $-R$ pada sumbu x ke titik R dimana $R = 2P$ dengan $R = (x_2, y_2)$, maka:

$$x_2 = m^2 - 2x_1 \; mod \; p \qquad (5)$$

$$y_2 = m(x_1 - x_3) - y_1 \; mod \; p \qquad (6)$$

Dimana gradien (m):

$$m = \frac{3{x_1}^2 + a}{2y_1} \qquad (7)$$

Dimana (x_1, y_1) koordinat titik asal dan (x_2, y_2) koordinat hasil perhitungan (Boruah & Saikia, 2014).

2.2.1.3 Perkalian Titik

Perkalian titik skalar merupakan blok dari semua kriptosistem kurva eliptik dengan Operasi dari bentuk kP, dimana P adalah titik pada kurva eliptik dan k adalah bilangan bulat positif. Menghitung kP berarti menambahkan titik P tepat $k - 1$ kali untuk diriya sendiri, yang menghasilkan titik lain Q pada kurva eliptik.

Perkalian titik menggunakan dua operasi dasar kurva eliptik diantaranya penjumlahan titik dan pengandaan titik. Misalnya menghitung $kP = Q$, jika $kP = 23P$ maka $kP = 23P = 2(2(2(2P) + P) + P) + P$, jadi untuk mendapatkan hasilnya, penjumlahan titik dan perkalian titik akan digunakan secara berulangkali (Boruah & Saikia, 2014).

2.2.2 Enkripsi dan Dekripsi Algoritma *Elliptic Curve Cryptography*

Dalam persoalan logaritma diskrit dari kurva eliptik, diberikan P dan Q yang merupakan dua buah titik di kurva eliptik, carilah integer k sedemikian sehingga Q = kP. Secara komputasi sulit untuk menemukan k, jika k adalah bilangan yang besar. Bilangan k merupakan logaritma diskrit dari Q dengan basis P. Pada ECC, Q adalah kunci publik, k adalah kunci privat, dan P adalah sembarang titik pada kurva eliptik.

Dalam kriptografi kunci asimetris, harus ditentukan terlebih dahulu nilai parameter yang akan digunakan dan telah disepakati oleh pihak yang akan berkomunikasi. Parameter yang digunakan dalam ECC yaitu nilai a dan b, bilangan prima p dalam persamaan kurva eliptik bidang terbatas serta titik generator G yang dipilih dari kurva eliptik. Pendekatan enkripsi dan dekripsi dengan ECC ini dapat dijelaskan dalam contoh kasus misalnya Alice ingin mengirim pesan yang terenkripsi kepada Bob berikut ini (Nugroho & Munir, 2015) :

1. Pembangkitan Kunci Privat dan Kunci Publik

   Bob membangkitkan kunci privat n_B dengan cara memilih bilangan acak yang nilainya diantara $[1, p-1]$. Dengan kunci privat tersebut, Bob membangkitkan kunci publik $P_B = n_B \, . \, G$.

2. Enkripsi

   Misalnya pesan yang akan dikirim adalah pesan m. Alice meng-encode pesan m menjadi integer nilai ASCII. Lalu memilih bilangan acak k yang nilai diantara $[1, p-1]$, kemudian dinyatakan x = (m.k) + 1, lalu sulihkan x kedalam persamaan kurva eliptik hingga mendapatkan nilai y, sehingga didapatkan pesan (Pm) telah menjadi

titik (x,y). Terakhir, Alice menghasilkan cipherteks (*Cm)*, yang terdiri dari pasangan titik $Cm = \{(kG), (Pm + kPB)\}$ dimana G adalah titik generator dan P_B adalah kunci publik Bob.

3. Dekripsi

   Untuk melakukan dekripsi cipherteks *Cm*, Bob mula-mula mengalikan titik pertama dari cipherteks dengan kunci privatnya n_B dan kemudian mengurangkan titik kedua dari cipherteks dengan hasil perkalian tersebut.

   $$Pm + kPB\ nB(kG) = Pm + k(nBG)\ nB(kG) = Pm$$

   Terakhir decode Pm (m = (x-1/n)) menjadi pesan m semula.

2.3 Chatting

Chatting ialah kegiatan berkomunikasi dalam dunia internet atau suatu fiture dalam internet untuk berkomunikasi langsung sesama pemakai internet yang sedang online. Anda dapat mengirimkan pesan kepada orang lain yang sedang online kemudian orang yang dituju akan merespon dengan membalas pesan Anda, demikian seterusnya. *Chatting* menggunakan metode komunikasi dua arah dimana data yang ditransmisikan atau dikirim dapat dilakukan secara dua arah dan dapat saling mengirimkan data secara bersama-sama. Bentuk komunikasi saat *chatting* dapat berupa suara, teks atau dalam bentuk video langsung dan berbicara tanpa teks. *Chatting* dapat terjadi di dalam percakapan umum maupun langsung melalui pesan pribadi. Pada dasarnya proses *chatting* membutuhkan sebuah aplikasi yang memfasilitasi terjadinya komunikasi *chatting*, dimana saat ini telah banyak aplikasi-aplikasi *chatting* yang bermunculan sesuai dengan kelebihan dan kekurangan masing-masing yang telah memberikan fasilitas pelayanan kepada setiap penggunanya (Fahlevie, 2012).

2.4 Aplikasi *Mobile*

Aplikasi *Mobile* adalah proses dimana pengembangan aplikasi untuk perangkat genggam seperti telepon genggam atau PDA. Dengan menggunakan aplikasi mobile dapat mempermudah berbagai macam aktivitas seperti hiburan,

maupun pekerjaan (Priyantono, 2019). Melalui aplikasi mobile, pengguna juga dapat mengakses sejumlah informasi informasi penting menggunakan smartphone yang terkoneksi dengan layanan internet. Perangkat mobile memiliki banyak jenis dalam ukuran desain dan tampilan namun memiliki karakteristik yang berbeda dengan desktop system. Selain itu perangkat mobile juga memiliki ukuran yang kecil serta memori yang terbatas. Selama manufaktur aplikasi mobile sudah ada atau bisa didownload oleh pemakai sesuai dengan platform perangkat lunaknya (Adha Bilqis Ibrahim & Gustina, 2021). Secara umum, aplikasi mobile memungkinkan penggunanya terhubung ke layanan internet yang biasanya hanya diakses melaului PC atau Notebook. Dengan demikian, aplikasi mobile dapat membantu pengguna untuk lebih mudah mengakses layanan internet menggunakan perangkat mobile mereka (Kadi, 2017).

2.5 Android

Android adalah sebuah sistem operasi pada handphone yang berbasis pada sistem operasi Linux dan bersifat terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Salah satu keunggulan Android yaitu sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat dan menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan *tools* yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat (Priyantono, 2019).

2.6 Flutter

Menurut (Adha Bilqis Ibrahim & Gustina, 2021) Flutter adalah kerangka antarmuka pengguna portabel (UI) Google untuk membangun aplikasi modern, asli, dan reaktif untuk berbagai platform. Flutter menggunakan *widget* untuk membuat UI. Flutter menggunakan mesin rendernya sendiri untuk menggambar *widget*. Elemen memiliki referensi ke *widget* dan bertanggung jawab untuk membandingkan perbedaan *widget*. Selain itu Flutter juga salah satu Software Development Kit (SDK) buatan Google yang berfungsi untuk membuat aplikasi mobile, baik untuk Android maupun iOS .Dengan Flutter, aplikasi Android dan iOS dapat dibuat menggunakan basis kode dan Bahasa pemrogaman yang sama yaitu Dart, bahasa pemrograman yang juga diproduksi oleh Google pada tahun 2011 (Luthfi, 2020). Salah satu keunggulan Flutter ialah semua kodenya di *compile* dalam kode *native* nya (Android NDK, LLVM, AOT-compiled) tanpa ada intrepeter pada prosesnya sehingga proses *compile*-nya menjadi lebih cepat.

2.7 Dart

Dart adalah sebuah bahasa pemrograman yang dikembangkan oleh Google, dirancang oleh Lars Bak dan Kasper Lund. Dart pertama kali dikenalkan pada 10 Oktober 2011 (Luthfi, 2020). Bahasa pemrograman ini dikembangkan sebagai bahasa pemrograman aplikasi yang dapat dengan mudah untuk dipelajari dan juga merupakan bahasa pemrograman berbasis *class* dan berorientasi terhadap obyek dengan menggunakan sitaks bahasa pemrograman. Dart dapat digunakan untuk membuat aplikasi server (berbentuk *commandline interface*), web, desktop, maupun mobile (Android dan iOS). Bahasa pemrograman Dart dapat digunakan secara bebas oleh para developer, karena bahasa ini dirilis secara *open-source* oleh Google di bawah lisensi BSD (Adha Bilqis Ibrahim & Gustina, 2021).

2.8 Firebase

Firebase adalah *Cloud Service Provider* dan *Backend as a Service* yang dimiliki oleh google. Firebase merupakan solusi yang ditawarkan oleh Google

untuk mempermudah dalam pengembangan aplikasi mobile maupun web. Firebase memiliki produk utama, yaitu menyediakan *backend* sebagai layanan *(Backend as a Service)* dan *database* realtime yang membuat data tetap terhubung di aplikasi klien melalui *listener realtime* dan menawarkan dukungan secara *offline* untuk seluler dan web sehingga dengan begitu dapat dibuat aplikasi yang responsif dan mampu bekerja tanpa harus bergantung pada latensi jaringan atau koneksi Internet. Layanan ini menyediakan pengembang aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di *cloud* Firebase ini (Adha Bilqis Ibrahim & Gustina, 2021).

2.9 Android Studio (IDE)

Menurut (Adha Bilqis Ibrahim & Gustina, 2021) Android Studio merupakan Integrated Development Enviroment (IDE) untuk system operasi Android, yang dibangung diatas perangkat lunak JetBrains IntelliJ IDEA dan didesain khusus untuk pengembangan aplikasi Android. IDE ini pertama kali diperkenalkan oleh Google dan diumumkan pada Mei 2013. Android studio memiliki beberapa keungulan diantaranya :

1. Editor kode yang cerdas, android studio membantu untuk membuat kode dengan cepat, dengan fitur intelligent code editor yang memberikan kemudahan dalam menganalisis kode dan menyediakan saran kode yang akan digunakan dengan sistem auto complete.
2. Android studio menyediakan layout editor yang lebih bagus dan Bisa melakukan build pada beberapa APK.
3. Dioptimalkan untuk seluruh perangkat android
4. Firebase assistant membantu menghubungkan aplikasi dengan Firebase dan menambahkan layanan seperti Analytics, Autentikasi, Notifikasi, dan lainnya dengan prosedur sesuai dengan urutan di dalam Android Studio.

2.10 Penelitian Terkait

Berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan :

Pada tahun 2021 penelitian oleh Danang H. Sulaksono, Citra N. Prabiantissa, Gusti E. Yuliastuti dan Ainur R. Taqwadari dari Jurusan Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya pada jurnal penelitian yang berjudul Implementasi Kriptografi dengan Metode *Elliptic Curve Cryptography* (ECC) untuk Aplikasi *Chatting* Berbasis Android. Penelitian ini membahas proses enkripsi (penyandian data) dan proses dekripsi (pengembalian data asli) pesan dengan algoritma *Elliptic Curve Cryptography* (ECC) pada aplikasi *chatting*. Parameter yang digunakan dalam mengukur efektivitas dari metode Algoritma *Elliptic Curve Cryptography* (ECC) yaitu *Avalanche Effect* yang akan menunjukkan bahwa suatu metode cocok digunakan untuk menyelesaikan masalah yang sedang terjadi saat ini. Hasil pengujian menunjukkan bahwa algoritma *Elliptic Curve Cryptography* (ECC) efektif untuk menyembunyikan *file* data pada aplikasi *chatting* yang bersifat privasi dengan hasil percobaan sebanyak 25 data citra didapatkan nilai *Avalanche Effect* terkecil adalah 26.528016 dan nilai *Avalanche Effect* terbesar adalah 94.67749 dengan rata – rata sebesar 79.88819. Hasil persentase yang cukup besar membuktikan bahwa aplikasi berjalan dengan baik, karena semakin besar nilai persentase yang didapat maka semakin baik aplikasi itu berjalan.

Pada tahun 2022 penelitian oleh Julieta Adhellia Pratiwi dan Asmunin dari Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya pada jurnal penelitian yang berjudul Penggunaan QR *code* Berbasis Kriptografi Menggunakan Algoritma *Elliptic Curve Criptography (ECC)*. Penelitian ini membahas proses enkripsi (penyandian data) dan proses dekripsi (pengembalian data asli) URL dengan algoritma *Elliptic Curve Cryptography* (ECC) pada QR Code. Kinerja yang diukur dari algoritma *Elliptic Curve Criptography (ECC)* ini, waktu komputasi yang dibutuhkan dalam melakukan enkripsi dan dekripsi data. Sebuah QR Code dengan implementasi algoritma *Elliptic Curve Criptography (ECC)* ini. Hasil Pengujian penerapan algoritma ECC digunakan untuk enkripsi dan dekripsi URL beberapa data penting dengan kecepatan dan performa *Elliptic Curve Criptography* diperoleh hasil rentan waktu rata-rata 0.00895 detik pada proses enkripsi dan waktu rata-rata

0.015878 detik pada proses dekripsi, menunjukkan semakin panjang masukkan diperoleh semakin banyak jumlah karakter semakin lama proses enkripsi dan dekripsi yang dibutuhkan.

Pada tahun 2021 penelitian oleh Ummu Wachidatul Latifah dan Puguh Wahyu Prasetyo dari Departemen Matematika, Fakultas Sains dan Teknologi Terapan dan Departemen Pendidikan Matematika, Fakultas Keguruan dan Ilmu Pendidikan, Universitas Ahmad Dahlan Yogyakarta pada jurnal penelitian yang berjudul Implementasi Kriptografi Kurva Eliptik Elgamal di Lapangan Galois Prima pada Proses Enkripsi dan Dekripsi Berbantuan *Software* Python. Penelitian ini membahas proses pembentukan kunci kurva eliptik, enkripsi (penyandian data) dan proses dekripsi (pengembalian data asli) dengan mengimplementasikan algoritma Kriptografi Kurva Eliptik Elgamal di Galois Field prima dengan menggunakan *software* Python. Hasil implementasi Kriptografi Kurva Eliptik ElGamal pada Galois Field prima menghasilkan sistem yang aman untuk menjaga kerahasiaan sebuah pesan dengan perhitungan titik – titik kurva eliptik yang rumit, sehingga hal tersebut sangat sulit diretas keamanannya. Hasil pembentukan kunci diperoleh kunci publik dan kunci privat. Input dari proses enkripsi berupa plaintext yang dienkripsi menggunakan kunci publik dan pada proses dekripsi memiliki input berupa ciphertext dan kunci privat dengan output berupa cipher text.

Pada tahun 2019 penelitian oleh Putri S E A Damanik dari jurusan Teknik Informatika, STMIK Budi Darma pada jurnal penelitian yang berjudul Implementasi Algoritma *Elliptic Curve Cryptography* (ECC) Untuk Penyandian Pesan Pada Aplikasi *Chatting Client Server* Berbasis Desktop. Penelitian ini membahas proses pembentukan kunci dan proses enkripsi (penyandian data) pesan teks dengan algoritma *Elliptic Curve Cryptography* (ECC) pada aplikasi *chatting client server* berbasis dekstop. Sebuah aplikasi yang dapat meng-enkripsi dan dekripsi sebuah pesan teks dengan implementasi algoritma *Elliptic Curve Criptography (ECC)*. Didapatkan hasil dengan konsep pengiriman pesan berdasarkan jaringan *peer to peer* dimana *client* dan *server* terhubung dalam jaringan untuk dapat melakukan pengiriman pesan dengan penerapan algoritma *Elliptic Curve Criptography (ECC)* dapat mengamankan

pesan teks, serta dapat menyajikan enkripsi dan dekripsi pesan teks dengan tepat.

Pada tahun 2020 penelitian oleh Heri Santoso, Mhd. dan Zulfansyuri Siambaton dari Univeristas Islam Negeri Sumatera Utara pada jurnal penelitian yang berjudul Aplikasi Pengamanan Ekstensi File Menggunakan Kriptografi One Time Pad (OTP) dan *Elliptic Curve Cryptography* (ECC). Penelitian ini membahas proses enkripsi (penyandian data) dan proses dekripsi (pengembalian data asli) file dengan algoritma *Elliptic Curve Cryptography* (ECC) dan One Time Pad (OTP) yang diterapkan pada sistem. Sebuah sistem yang dapat meng-enkripsi dan dekripsi sebuah file dengan implementasi algoritma *Elliptic Curve Criptography (ECC)* dan One Time Pad (OTP). Didapatkan hasil proses enkripsi *plaintext* menggunakan algoritma *One Time Pad* dapat melindungi informasi yang terdapat dalam file teks tersebut. Proses enkripsi menggunakan *One Time Pad* adalah jumlah karakter kunci harus sepanjang karakter *plaintext*, sedangkan untuk proses enkripsi menggunakan Algoritma *Elliptic Curve Cryptography (ECC)* dapat melindungi informasi yang terdapat dalam *file* teks, dengan proses enkripsi untuk mengubah satu titik ke titik yang lain sebagai *ciphertext*, sedangkan untuk proses dekripsi mengubah satu titik (chipertext) ke titik semula. Namun pada penelitian ini algoritma *Elliptic Curve Cryptography* dilakukan pengujian enkripsi didapatkan hasil bahwa beberapa *file* seperti gambar dan video tidak dapat dienkripsi dengan baik.

BAB III
METODE PENELITIAN

Pada bab ini akan membahas mengenai analisis kebutuhan sistem, perancangan implementasi sistem, implementasi algoritma, skenario penggunaan dan skenario pengujian.

3.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem mencakup penentuan spesifikasi dari perangkat keras dan perangkat lunak yang akan digunakan.

3.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah satu buah laptop dengan spesifikasi dapat dilihat pada tabel 1.

Tabel 1. Spesifikasi Laptop

Processor	11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz (12 CPUs), ~2.7GHz
Ukuran RAM	8 GB
Ukuran SSD	512 GB

3.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sebagai berikut:

1. Sistem Operasi

   Sistem operasi yang digunakan pada komputer penelitian menggunakan Windows 11 Home Single Language 64 bit.

2. Android Studio

   Android Studio merupakan *Integrated Development Enviroment* (IDE) untuk pengembangan aplikasi *mobile* terkhususnya untuk sistem operasi Android. Android Studio yang digunakan pada penelitian ini adalah Android Studio Bumblebee versi 2021.1.1

3. Flutter

Flutter merupakan sebuah *framework* dalam mengembangkan aplikasi *multi-platform,* salah satunya ialah android. Flutter yang digunakan pada penelitian ini adalah versi 3.0.3

4. Dart

   Dart merupakan bahasa pemrograman yang digunakan dalam *framework* Flutter. Bahasa pemrograman yang digunakan pada penelitian ini adalah Dart versi 2.17.5 (*stable*)

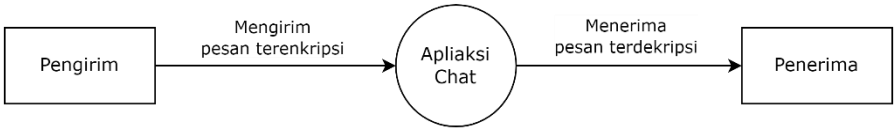
3.2 Perancangan Implementasi Sistem

Sebelum memulai pembuatan sistem, terlebih dahulu dilakukan analisis untuk meminimalkan kesalahan dalam pembuatan sistem, serta mengetahui apa saja yang diperlukan dalam pembuatan sistem. Diharapkan dengan adanya analisis sistem dapat membangun aplikasi yang tepat guna dan sesuai harapan.

Sistem yang dibangun merupakan aplikasi *chat* yang memiliki fungsi utama yaitu mengirim dan menerima pesan. Didalam aplikasi terdapat sistem kriptografi yang mencakup proses membangkitkan kunci publik dari masukan kunci privat, serta melakukan enkripsi dan dekripsi pesan. Setelah membangkitkan kunci publik, aplikasi akan menyimpan kunci publik di server. Dalam proses pengiriman pesan, aplikasi akan mengenkripsi pesan tersebut terlebih dahulu sebelum dikirim dengan menggunakan kunci publik penerima. Kunci publik penerima tersebut diperoleh dengan cara meminta ke server. Sedangkan ketika menerima pesan, aplikasi akan mendekripsi pesan terenkripsi yang diterimanya dengan kunci privat. Berikut ini rancangan sistem yang digambarkan dengan *Data Flow Diagram* (DFD) sebagai berikut:

3.2.1 *Context Diagram*

Context Diagram merupakan garis besar *operasional system*. *Context Diagram* menjelaskan secara sederhana dalam bentuk diagram dari proses kerja sistem. Rancangan diagram konteks aplikasi *chat* dapat dilihat pada gambar 7.

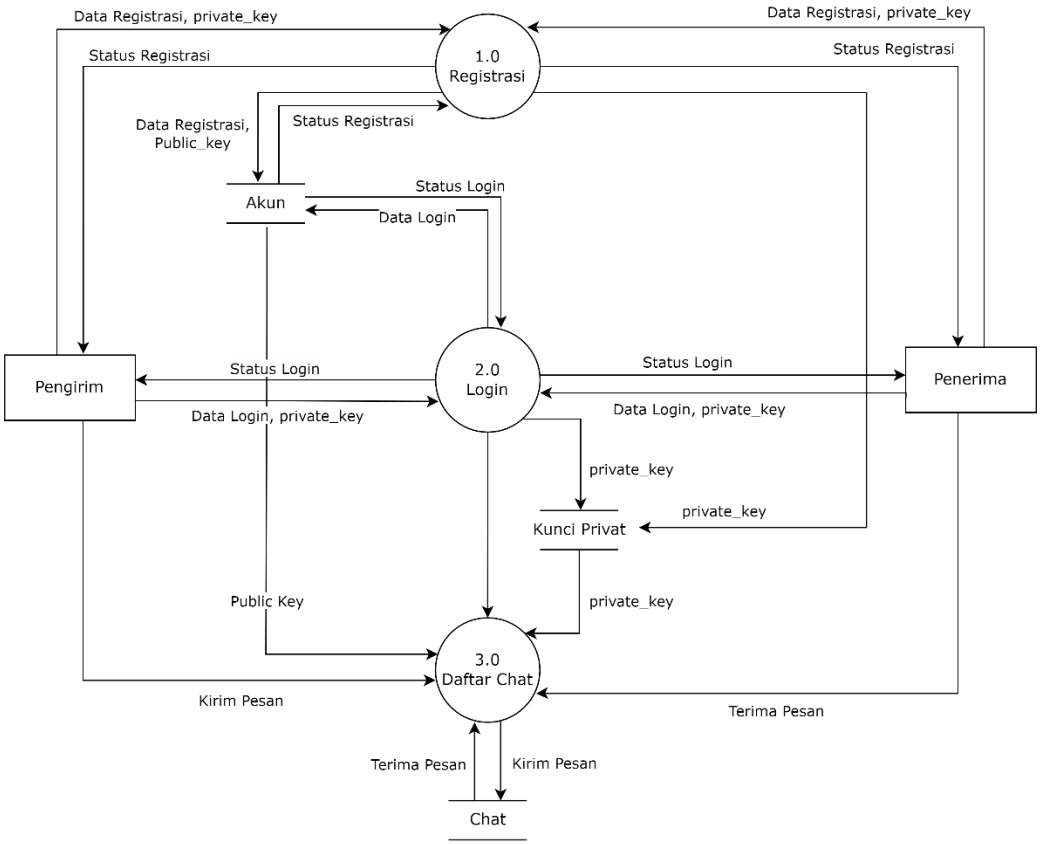


Gambar 7 *Context Diagram*

Dalam mengembangkan sistem ini, dibuat sebuah *Context Diagram* yang menjelaskan bahwa pengirim dapat mengirim pesan terenkripsi dengan menginputkan pesan berupa teks. Setelah diproses oleh sistem, penerima akan menerima pesan yang terenkripsi berubah menjadi pesan yang dapat dibaca.

3.2.2 *Data Flow Diagram* (DFD) *Level* 0

Data Flow Diagram Level 0 merupakan perincian dari *Context Diagram* yang menjelaskan lebih detail tentang yang terjadi didalam sistem. Tampilan *data flow diagram level* 0 dapat dilihat pada gambar 8.



Gambar 8 *Data Flow Diagram Level* 0

Dari gambar 8 dapat dijelaskan bahwasanya aplikasi ini memiliki 3 proses pokok untuk melayani pengguna dengan baik sebagai pengirim maupun penerima yaitu: registrasi, *login*, dan daftar *chat*. Pada proses daftar chat, pengguna (pengirim atau penerima) diharuskan untuk melakukan proses login terlebih dahulu untuk memvalidasi pengguna. Penjelasan DFD Level 0 dapat dilihat pada Tabel 2.

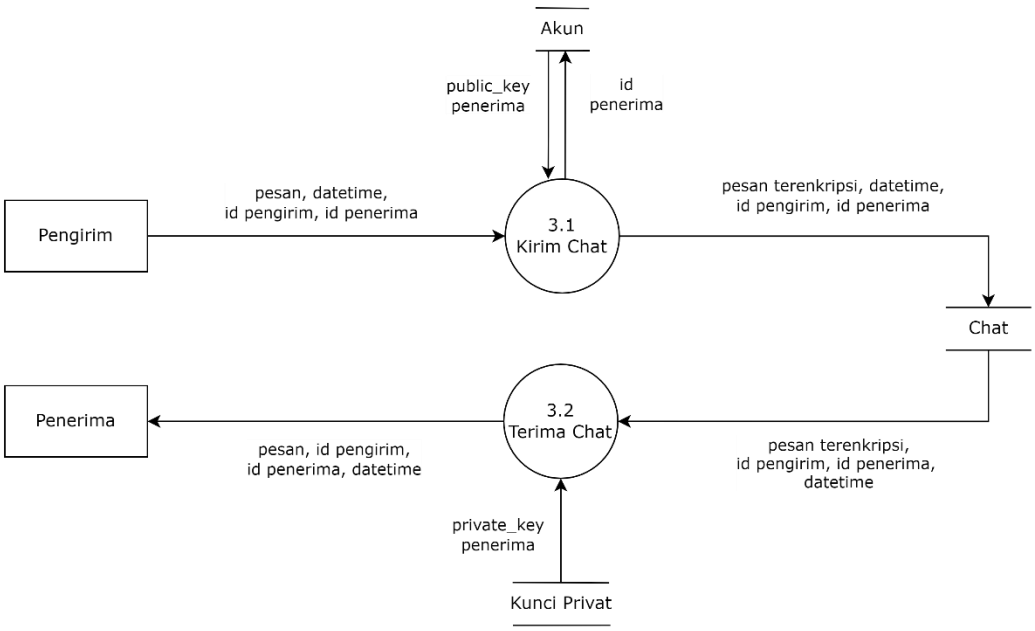
Tabel 2. Keterangan proses DFD *Level* 0

Nama Proses	**Deskripsi**
Registrasi (1.0)	Pada proses ini pengguna (pengirim dan penerima) melakukan penginputan Data Registrasi berupa *username*, nama, *email*, dan *password* (*private_key*). Pada proses Registrasi menghasilkan *public_key* dari inputan *private_key* dan id pengguna. Data Registrasi akan disimpan di *database* Akun seperti *username*, nama, *email*, id pengguna dan *public_key* yang dihasilkan dari inputan *private_key*, sedangkan untuk *password* yang akan bertindak sebagai *private_key* di simpan kedalam *database* Kunci Privat, lalu mengembalikan Status Registrasi ke proses Registrasi sampai ke pengguna (pengirim dan penerima).
Login (2.0)	Pada proses ini pengguna (pengirim dan penerima) melakukan penginputan Data *login* berupa *email* dan *password* yang akan bertindak sebagai *private_key*. Data *login* akan dilakukan pengecekan pada proses *login* dengan *database* Akun yang akan mengembalikan status *login*, kemudian status *login* akan dikirimkan ke

	pengguna (pengirim dan penerima). Pada proses *login*, inputan *private_key* akan disimpan di *database* lokal Kunci Privat.
Daftar *Chat* (3.0)	Pada proses ini pengirim dapat mengirimkan pesan dan penerima dapat menerima pesan. Pesan yang dikirim oleh pengirim akan disimpan di *database Chat*, sedangkan penerima akan menerima pesan yang diambil dari *database Chat*. Pada proses Daftar *Chat*, akan menerima *private_key* dari *database* lokal Kunci Privat.

3.2.3 *Data Flow Diagram* (DFD) Daftar *Chat Level* 1

Data Flow Diagram (DFD) *Level* 1 merupakan perincian yang lebih detail dari proses Daftar *Chat* yang ada pada DFD Level 0. DFD *Level* 1 untuk proses 3 yaitu proses Daftar *Chat* dapat dilihat pada gambar 9.



Gambar 9 *Data Flow Diagram* (DFD) Daftar *Chat Level* 1

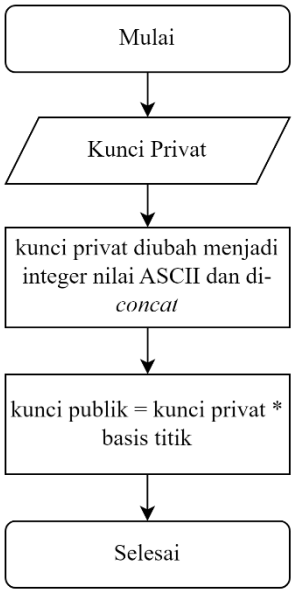
Pada gambar 9 terdapat dua proses utama yaitu proses Kirim *Chat* dan Terima *Chat*. Penjelasan *Data Flow Diagram* (DFD) Daftar *Chat Level* 1 dapat dilihat pada Tabel 3.

Tabel 3. Keterangan proses DFD Daftar *Chat Level* 1

Nama Proses	**Dekripsi**
Kirim *Chat* (3.1)	Pada proses ini, pengirim menginputkan pesan berupa teks yang akan dienkripsi. Kemudian pada *database* Akun, Kunci Publik akan menerima id penerima yang akan dicocokkan dengan id yang tersimpan di *database* Akun. Setelah mengetahui *public_key* dari masukan id penerima, *database* Akun akan mengembalikan *public_key* penerima. Kemudian dilakukan enkripsi pesan dengan menggunakan public_key penerima dan dikirim ke *database* Chat.
Terima *Chat* (3.2)	Pada proses ini akan menerima pesan dari *database Chat* dan mengirimkannya kepada penerima sesuai dengan id penerima dan id pengirim, dimana dalam proses Kirim *Chat* (3.1) dan proses Terima *Chat* (3.2) nantinya pesan akan diurutkan kepada pengirim dan penerima sesuai id pengirim dan id penerima yang tersimpan dalam *database Chat*. Pada proses Terima *Chat*, akan dilakukan proses dekripsi pesan terenkripsi yang diterima dari *database Chat* dengan *private_key* penerima yang tersimpan di *database* Kunci Privat.

3.3 Implementasi Algoritma

3.3.1 Proses Pembangkitan Kunci Publik

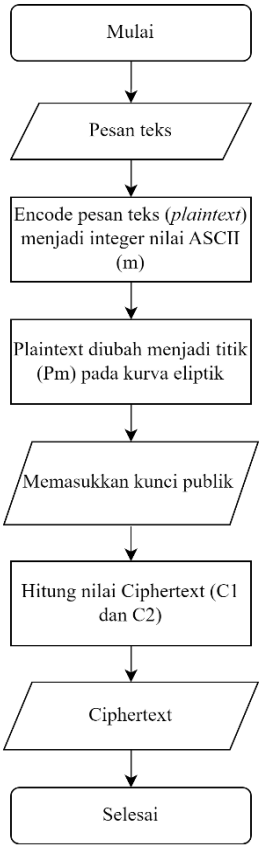


Gambar 10 Pembangkitan kunci publik *Elliptic Curve Cryptography*

Dari gambar 10 dapat dijelaskan langkah-langkah proses pembangkitan kunci publik sebagai berikut:

1. Masukkan kunci privat berupa teks *string* maupun angka.
2. *Encode* setiap karakter kunci privat menjadi integer nilai ASCII dan di-*concat* setiap karakter integer nilai ASCII.
3. Hitung hasil perkalian titik kunci privat dengan basis titik untuk mendapatkan kunci publik.

3.3.2 Proses Enkripsi

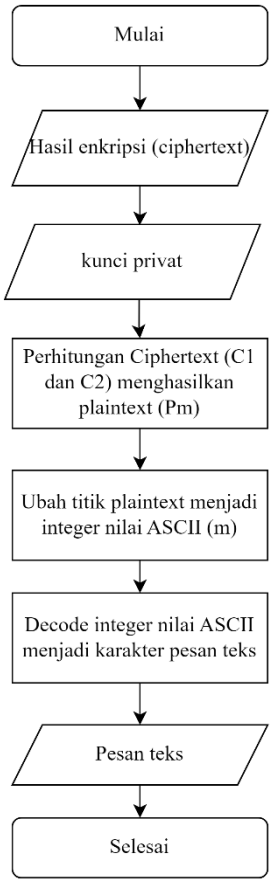


Gambar 11 Proses enkripsi *Elliptic Curve Cryptography*

Dari gambar 11 dapat dijelaskan langkah-langkah proses enkripsi pesan sebagai berikut:

1. Masukkan pesan teks berupa *string* maupun angka
2. *Encode* setiap karakter *plaintext* (m) yang diinput menjadi integer nilai ASCII.
3. *Plaintext* (m) yang telah bernilai integer ASCII, dilakukan perhitungan agar menjadi titik pada kurva eliptik (Pm).
4. Masukkan kunci publik (Pb) yang digunakan dalam mengenkripsi pesan.
5. Hitung nilai C1 (k.G) dan C2 (Pm + k.Pb) yang menghasilkan titik *ciphertext*, sehingga diperoleh *ciphertext* (C1, C2).

3.3.3 Proses Dekripsi



Gambar 12 Proses dekripsi *Elliptic Curve Cryptography*

Dari gambar 12 dapat dijelaskan langkah-langkah proses dekripsi pesan sebagai berikut:

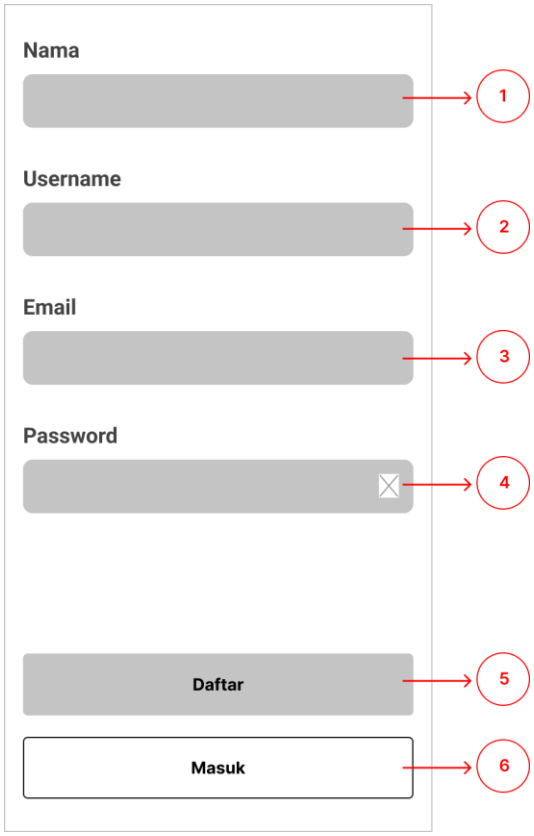
1. Masukkan hasil enkripsi pesan teks berupa *ciphertext*
2. Masukkan kunci privat (b) yang digunakan untuk mengdekripsi pesan.
3. Pada *ciphertext* dilakukan operasi perhitungan (C2 – b.C1) untuk mendapatkan kembali nilai titik *plaintext* (Pm).
4. Dilakukan operasi perhitugan pada titik *plaintext* (Pm) untuk mendapatkan kembali pesan yang nilai integer ASCII (m).
5. *Decode* pesan yang bernilai integer ASCII untuk mendapatkan karakter pesan teks seperti semula.

3.4 Perancangan Antar Muka Sistem

Perancangan antar muka (*interface*) merupakan rancangan yang menggambarkan interaksi pengguna dengan sistem dari segi tampilan atau *display*, serta kemudahan pengguna dalam pengoperasiannya.

A. Halaman Registrasi

Berikut ini adalah rancangan dari halaman registrasi :



Gambar 13 Halaman Registrasi

Pada gambar 13 dapat dilihat bahwa terdapat empat *input field* yang harus diisi dan dua *command button* yaitu: nama, *username*, *email*, *password*, daftar dan masuk.

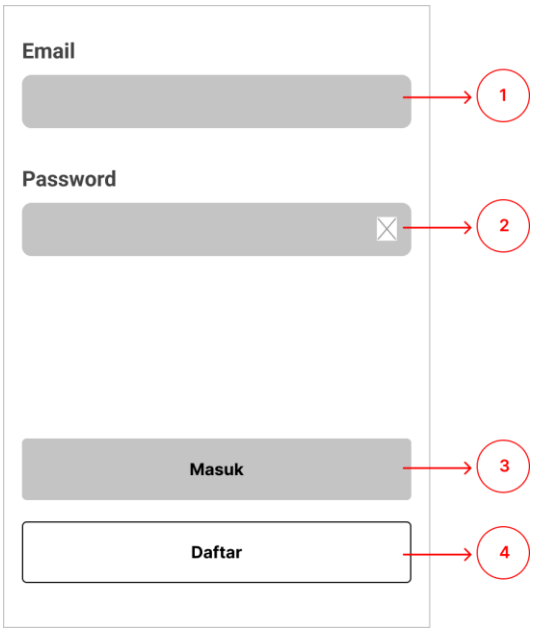
Keterangan:

1. Nama, nama pengguna yang akan menggunakan aplikasi *chat* ini.
2. *Username*, sebagai identitas pengguna yang unik antara satu pengguna dengan pengguna lainnya pada saat pencarian pengguna.

3. *Email*, sebagai identitas pengguna yang unik antara satu pengguna dengan pengguna lainnya pada saat aktifitas *login.*
4. *Password*, sebagai aktifitas *login* dan *private key* untuk mengdekripsi pesan yang masuk.
5. Daftar, tombol untuk mengirim informasi dari form registrasi kepada sistem untuk disimpan kedalam *database.*
6. Masuk, tombol untuk kembali kehalaman *login.*

B. Halaman *Login*

Rancangan dari halaman *login* dapat dilihat pada gambar 14.



Gambar 14 Halaman *Login*

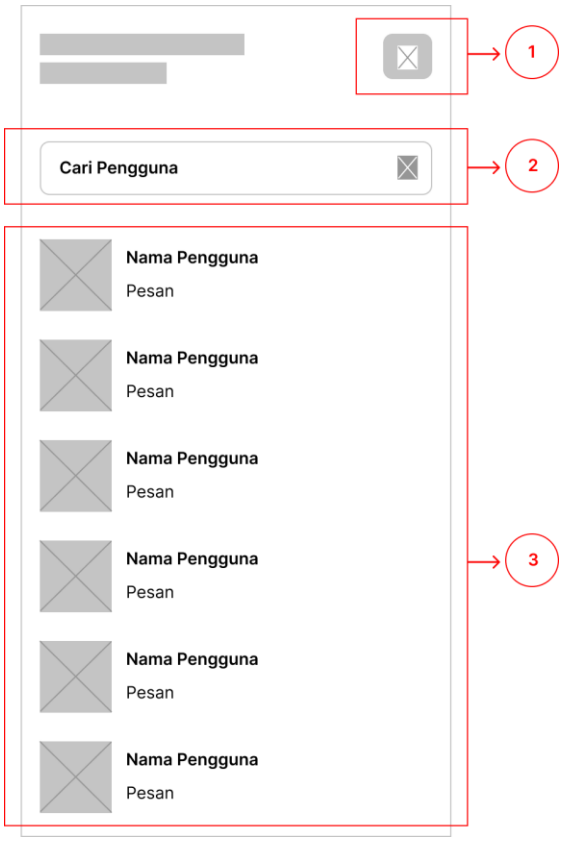
Pada gambar 14 dapat dilihat bahwa terdapat dua *input field* yang harus diisi dan dua *command button* yaitu: *email*, *password*, masuk dan daftar.

Keterangan :

1. *Username*, untuk melakukan autentikasi pengguna kedalam sistem.
2. *Password*, untuk melakukan autentikasi pengguna kedalam sistem.
3. Masuk, tombol untuk mengirim kombinasi *username* dan *password* kepada sistem untuk diproses.
4. Daftar, tombol untuk masuk kehalaman registrasi

C. Daftar *Chat*

Rancangan dari Halaman Daftar *Chat* dapat dilihat pada gambar 15.



Gambar 15 Halaman Daftar *Chat*

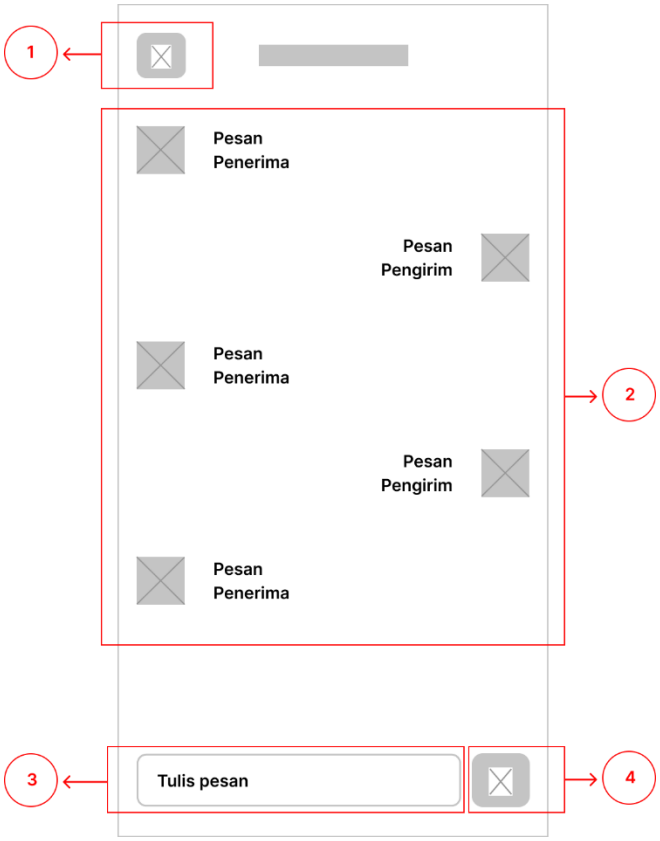
Pada gambar 15 dapat dilihat bahwa terdapat satu *input field*, satu *command button* dan satu tampilan *list* yaitu: *search user*, *navigation drawer button*, dan *list chat*.

Keterangan :

1. *Navigation drawer button*, untuk melakukan pemilihan opsi berupa *logout* akun pengguna kepada sistem.
2. *Search user*, untuk melakukan pencarian pengguna menggunakan inputan *username* pengguna.
3. *List Chat*, daftar *chat* pengguna dengan pengguna lainnya yang telah mengirim atau menerima pesan.

D. Halaman *Chat*

Rancangan dari Halaman *Chat* dapat dilihat pada gambar 16.



Gambar 16 Halaman *Chat*

Pada gambar 16 dapat dilihat bahwa terdapat satu *input field*, dua *command button* dan satu tampilan *list* yaitu: *input* pesan, *back button*, *send button* dan *list* pesan.

Keterangan :

1. *Back button*, tombol kembali untuk kembali kehalaman sebelumnya.
2. *List* Pesan, daftar pesan antar pengguna dengan penerima pesan.
3. *Input* pesan, untuk menuliskan pesan yang akan dikirim kepengguna.
4. *Send button*, tombol kirim untuk mengirim pesan yang telah ditulis pada *input* pesan.

3.5 Skenario Penggunaan Sistem

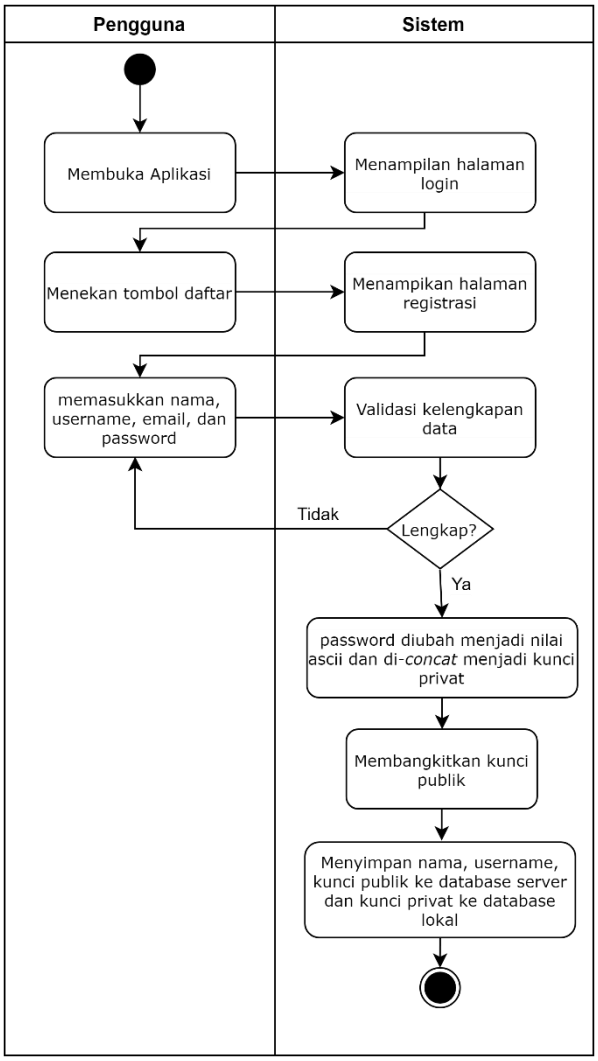
Activity diagram digunakan dalam mengsimulasikan skenario penggunaan sistem untuk menampilkan secara detail dari penggunaan sistem. Pada sistem ini, pengguna sistem meliputi pengirim dan penerima pesan yang

akan melakukan pembangkitan kunci publik, enkripsi, dan dekripsi pesan. Berikut tugas antara pengirim dan penerima:

1. Pengguna (Pengirim dan penerima) akan melewati proses registrasi untuk membangkitkan kunci publik.
2. Pengirim, bertindak sebagai pengguna yang akan melakukan enkripsi terhadap pesan yang akan dikirim.
3. Penerima, bertindak sebagai pengguna yang akan melakukan dekripsi terhadap pesan yang diterima.

3.5.1 *Activity Diagram* Proses Registrasi dan Pembangkitan Kunci Publik

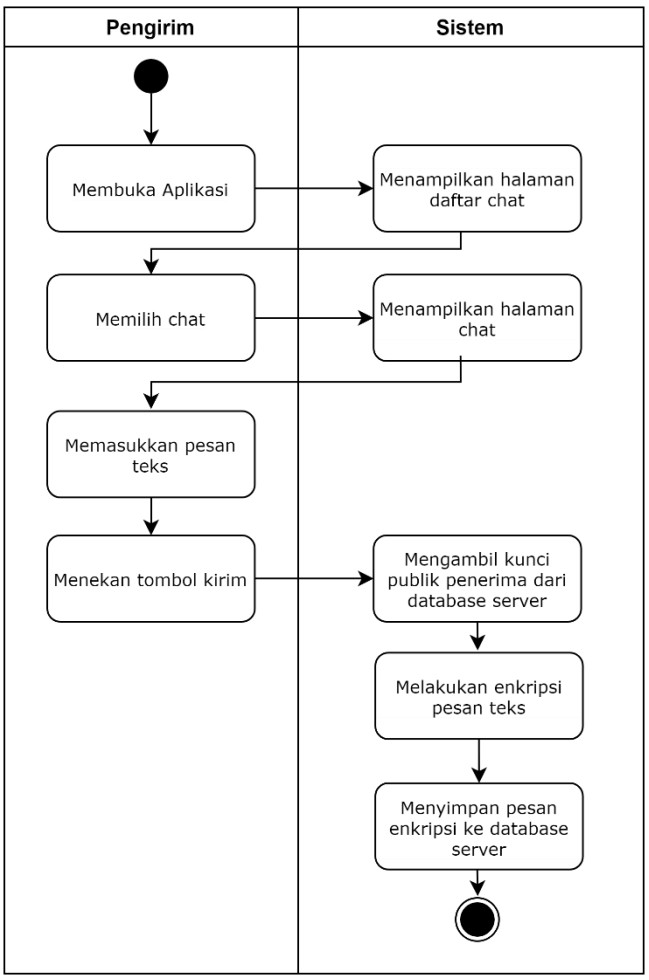
Activity diagram proses registrasi dan pembangkitan kunci publik berfungsi untuk menampilkan pengguna melakukan proses registrasi dan tanggapan sistem mengenai permintaan pengguna. *activity diagram* dari proses registrasi dan pembangkitan kunci publik dapat dilihat pada gambar 17.



Gambar 17 *Activity diagram* registrasi dan pembangkitan kunci publik

3.5.2 *Activity Diagram* Proses Enkripsi dan Kirim Pesan

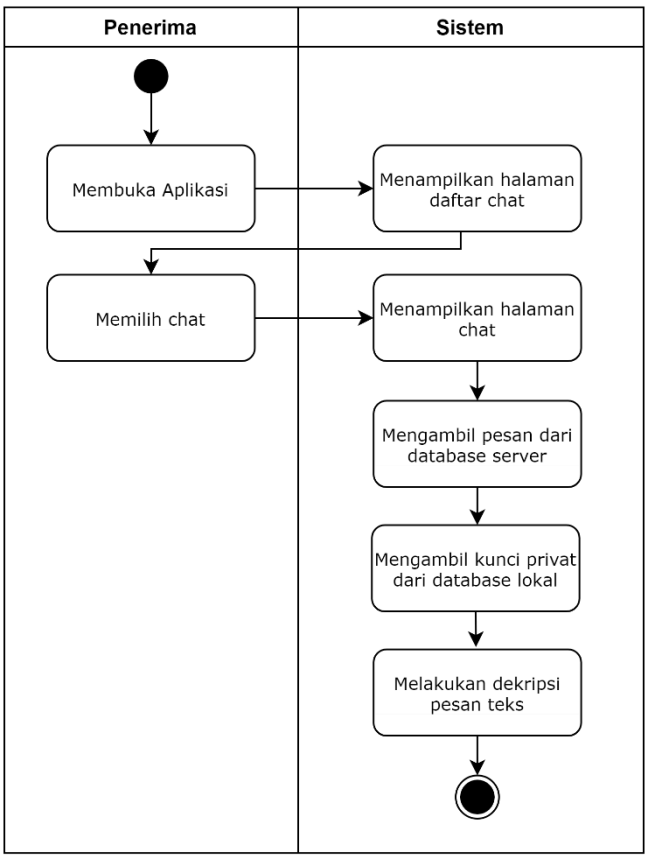
Activity diagram proses enkripsi dan kirim pesan berfungsi untuk menampilkan proses pengirim melakukan aktivitas pengiriman pesan dengan sistem dan tanggapan sistem mengenai permintaan pengirim. *activity diagram* dari proses enkripsi dan kirim pesan dapat dilihat pada gambar 18.



Gambar 18 *Activity diagram* proses enkripsi dan kirim pesan

3.5.3 *Activity Diagram* Proses Dekripsi dan Terima Pesan

Activity diagram proses dekripsi dan terima pesan berfungsi untuk menampilkan proses penerima melakukan aktivitas penerimaan pesan dengan sistem dan tanggapan sistem mengenai permintaan penerima. *activity diagram* dari proses dekripsi dan terima pesan dapat dilihat pada gambar 19.



Gambar 19 *Activity diagram* proses dekripsi dan terima pesan

3.6 Skenario Pengujian

Pengujian pada sistem terbagi menjadi 3 bagian, yaitu pengujian fungsionalitas aplikasi menggunakan metode *black box*. Pengujian implementasi *Elliptic Curve Cryptography* pada aplikasi, dan pengujian algoritma pada proses enkripsi dan dekripsi teks.

3.6.1 Pengujian *Black Box*

Pengujian *blackbox* (*blackbox testing)* adalah salah satu metode pengujian perangkat lunak yang berfokus dengan melihat fungsi – fungsi yang ada dalam sistem tanpa harus mengetahui bagaimana fungsi tersebut dibuat sistemnya. Tahap pengujian atau testing merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak. Tabel skenario pengujian *black box* dapat dilihat pada tabel 4.

Tabel 4. Skenario pengujian *Black Box*

Pengujian Aplikasi		
No.	***Test Case***	**Skenario Pengujian**
1.	Halaman *Login*	Mengisi dengan benar data pengguna berupa *email* dan *password*, kemudian menekan tombol masuk.
		Mengisi data *email* dan *password* pengguna dengan salah, kemudian menekan tombol masuk.
		Menekan tombol register.
2.	Halaman Register	Mengisi data nama, *email*, *username* dan *password*, kemudian menekan tombol daftar.
		Menekan tombol masuk.
3.	Halaman Daftar *Chat*	Menekan salah satu *chat* pengguna.
		Mengisi *username* pengguna kemudian menekan tombol pencarian.
4.	Halaman *Chat*	Menekan tombol kembali.
		Mengisi pesan teks dan menekan tombol kirim.

3.6.2 Pengujian Implementasi *Elliptic Curve Cryptography* pada Aplikasi

Pada pengujian implementasi *Elliptic Curve Cryptography* pada aplikasi merupakan pengujian sistem yang telah diterapkannya kriptografi algoritma *Elliptic Curve Cryptography*, dimana akan menguji hasil pembangkitan kunci publik pada proses registrasi, enkripsi pada proses pengiriman pesan dan dekripsi pada proses penerimaan pesan.

Pengujian *Elliptic Curve Cryptography* pada aplikasi *chat* dengan *end-to-end encryption* juga akan memantau *request* yang dilakukan ke *server* menggunakan *firebase emulator* untuk mengetahui apakah pesan

yang dikirimkan sudah dalam bentuk *ciphertext* atau tidak, maka pengujian ini akan dibagi menjadi 2 macam, yaitu ketika fungsi enkripsi diaktifkan dan tidak diaktifkan.

3.6.3 Pengujian Enkripsi dan Dekripsi *Elliptic Curve Cryptography*

Pada pengujian ini, algoritma *Elliptic Curve Cryptography* akan dilakukan pengujian terhadap enkripsi dan dekripsi dari parameter standar kriptografi kurva eliptik Secp192r1, Secp256r1, dan Secp521r1 (Hammi dkk., 2020). Pengujian dilakukan sebanyak 15 kali pada proses enkripsi dan dekripsi dengan panjang teks yang sama dan berbeda. Hasil yang didapatkan dari pengujian proses enkripsi dan dekripsi dengan panjang teks yang sama dilakukan perhitungan waktu rata-rata (mean) dan standar deviasi (simpangan baku) dari variasi hasil data uji tersebut. Selain itu dilakukan juga perbandingan kunci publik dengan *ciphertext* yang dihasilkan dari proses enkripsi dengan panjang teks yang berbeda. Spesifikasi parameter kurva eliptik yang diuji dengan bentuk heksadesimal pada nilai variabelnya dapat dilihat pada tabel 5, 6, dan 7.

Tabel 5. Parameter Secp192r1

Secp192r1	
Parameter	Nilai variabel
p	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFF
a	FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFC
b	64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
G	(188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012, 07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811)

Tabel 6. Parameter Secp256r1

Secp256r1	
Parameter	Nilai variabel
p	FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFF
a	FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFC
b	5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E 27D2604B
G	(6B17D1F2 E12C4247 F8BCE6E5 63A440F2 77037D81 2DEB33A0 F4A13945 D898C296, 4FE342E2 FE1A7F9B 8EE7EB4A 7C0F9E16 2BCE3357 6B315ECE CBB64068 37BF51F5)

Tabel 7. Parameter Secp521r1

Secp521r1	
Parameter	Nilai variabel
p	01FFFFFF FFFF
a	01FFFFFF FFFC
b	0051953E B9618E1C 9A1F929A 21A0B685 40EEA2DA 725B99B3 15F3B8B4 89918EF1 09E15619 3951EC7E 937B1652 C0BD3BB1 BF073573 DF883D2C 34F1EF45 1FD46B50 3F00

G	(00C6858E 06B70404 E9CD9E3E CB662395 B4429C64 8139053F B521F828 AF606B4D 3DBAA14B 5E77EFE7 5928FE1D C127A2FF A8DE3348 B3C1856A 429BF97E 7E31C2E5 BD66, 01183929 6A789A3B C0045C8A 5FB42C7D 1BD998F5 4449579B 446817AF BD17273E 662C97EE 72995EF4 2640C550 B9013FAD 0761353C 7086A272 C24088BE 94769FD1 6650)

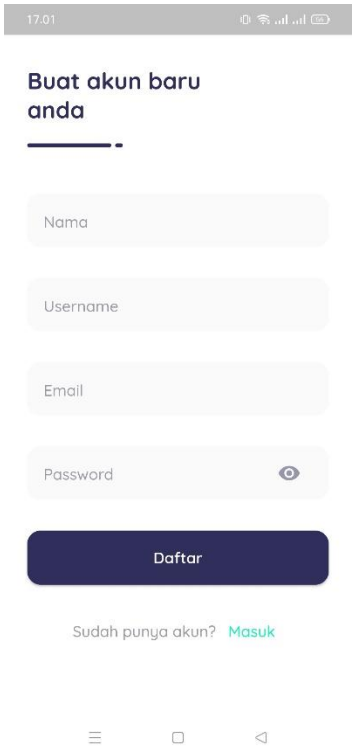
BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi Antar Muka Aplikasi *Chat*

Berdasarkan rancangan dari tampilan antar muka aplikasi pada bab sebelumnya, berikut ini adalah beberapa hasil rancangan antar muka aplikasi *chat* yang dirancang dengan *Framework* Flutter dengan Bahasa Pemrograman Dart.

A. Halaman Registrasi

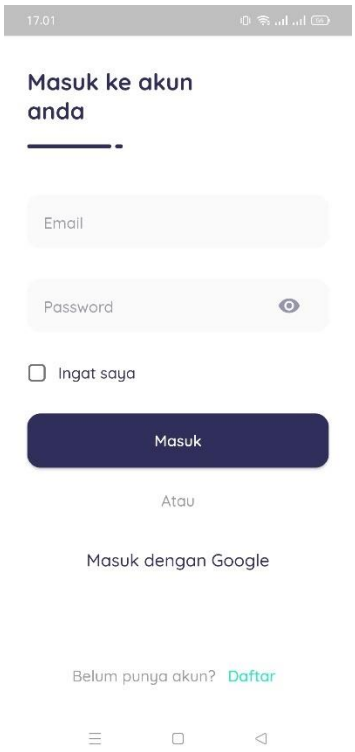
Registrasi, fungsi register mendaftarkan pengguna baru ke server serta membuat kunci publik untuk pengguna dan menjadikan password pengguna sebagai kunci privat. Kunci publik di-*upload* ke *database server*, sedangkan kunci privat disimpan ke *database* lokal. Tampilan dari halaman registrasi dapat dilihat pada gambar 20.



Gambar 20 Halaman Registrasi

B. Halaman *Login*

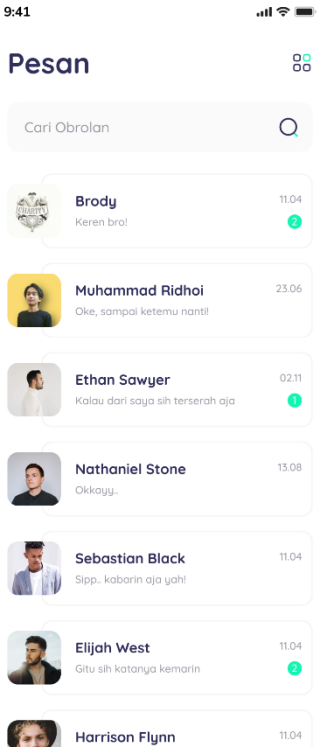
Login, fungsi *login* menghubungkan pengguna dengan *server* setelah mendaftarkan data diri menggunakan *email* dan *password* yang telah didaftarkan sebelumnya. Tampilan dari halaman *login* dapat dilihat pada gambar 21.



Gambar 21 Halaman *Login*

C. Halaman Daftar *Chat*

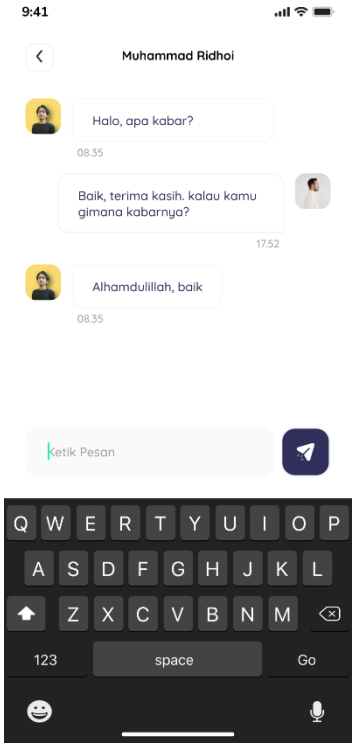
Halaman Daftar *Chat*, menampilkan daftar *chat* dengan pengguna lain yang telah berkirim pesan pada pengguna, serta dapat mencari pengguna lain yang telah terdaftar di *server* pada fungsi pencarian pengguna. Tampilan dari halaman daftar *chat* dapat dilihat pada gambar 22.



Gambar 22 Halaman Daftar *Chat*

D. Halaman *Chat*

Halaman *Chat*, berfungsi mengirimkan pesan ke pengguna lain dan menampilkan pesan yang dikirim antara dua pengguna. Tampilan dari halaman *chat* dapat dilihat pada gambar 23.



Gambar 23 Halaman *Chat*

4.2 Pengujian *Black Box*

Pengujian *Black Box* berfungsi untuk melihat fungsi – fungsi yang ada dalam sistem tanpa harus mengetahui bagaimana fungsi tersebut dibuat sistemnya. Hasil dari pengujian *Black Box* dapat dilihat pada tabel 8.

Tabel 8. Pengujian *Black Box*

Pengujian Aplikasi				
No.	***Test Case***	**Skenario Pengujian**	**Hasil yang Diharapkan**	**Kesimpulan**
1.	Halaman *Login*	Mengisi dengan benar data pengguna berupa *email* dan *password*, kemudian menekan tombol masuk.	Sistem akan menampilkan halaman daftar *chat*	Valid

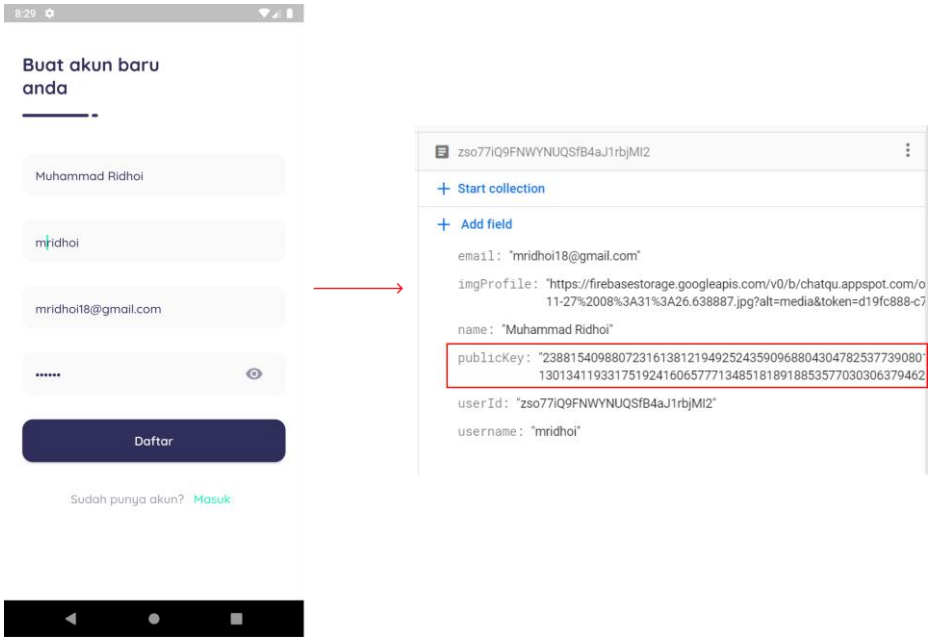
		Mengisi data *email* dan *password* pengguna dengan salah, kemudian menekan tombol masuk.	Sistem akan menolak dan menampilkan pesan email atau password salah.	Valid
		Menekan tombol register.	Sistem akan menampilkan halaman registrasi	Valid
2.	Halaman Registrasi	Mengisi data nama, *email, username* dan *password,* kemudian menekan tombol daftar.	Sistem akan menyimpan data pengguna kemudian menampilkan halaman daftar *chat*	Valid
		Menekan tombol masuk.	Sistem akan menampilkan halaman *login*	Valid
3.	Halaman Daftar *Chat*	Menekan salah satu *chat* pengguna.	Sistem akan menampikan halaman *chat* berdasarkan *chat* yang dipilih.	Valid
		Mengisi *username* pengguna kemudian menekan tombol pencarian.	Sistem akan menampilkan pengguna berdasarkan *username* yang dicari.	Valid
4.	Halaman *Chat*	Menekan tombol kembali.	Sistem akan menampilkan	Valid

			halaman daftar *chat*	
		Mengisi pesan teks dan menekan tombol kirim.	Sistem akan menampilkan pesan teks yang dikirim pada halaman *chat*.	Valid

4.3 Pengujian Implementasi *Elliptic Curve Cryptography* pada Aplikasi

4.3.1 Pembangkitan Kunci Publik

Kunci publik akan dibangkitkan melalui proses pendaftaran, kemudian hasil dari proses pendaftaran tersebut menghasilkan kunci publik yang dikirim ke server, hasil percobaan registrasi dapat dilihat pada gambar 24.

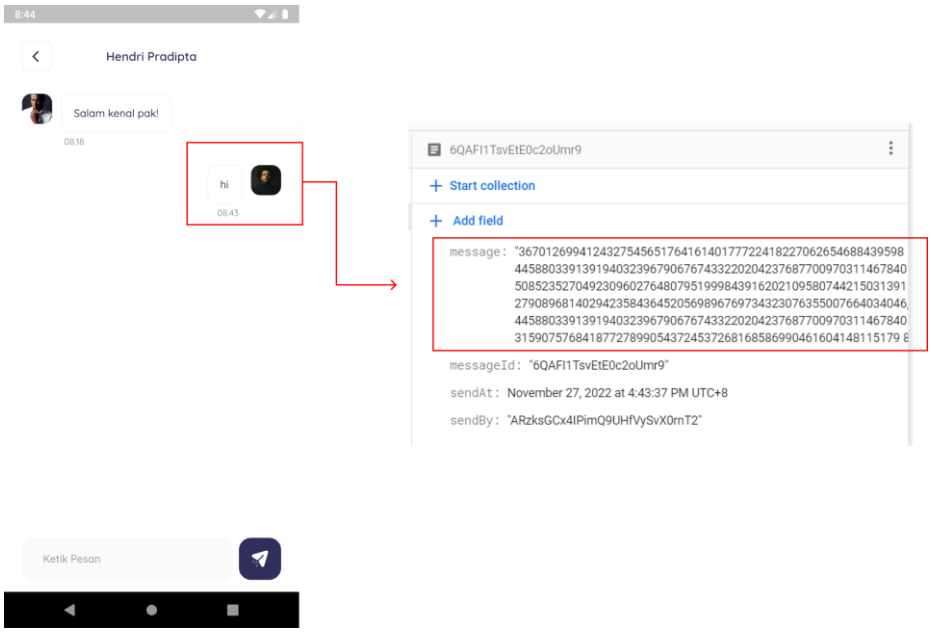


Gambar 24 Pembangkitan kunci publik

4.3.2 Enkripsi Pesan

Enkripsi pesan akan dilakukan melalui proses pengiriman pesan, kemudian hasil dari enkripsi pesan tersebut menghasilkan *ciphertext*

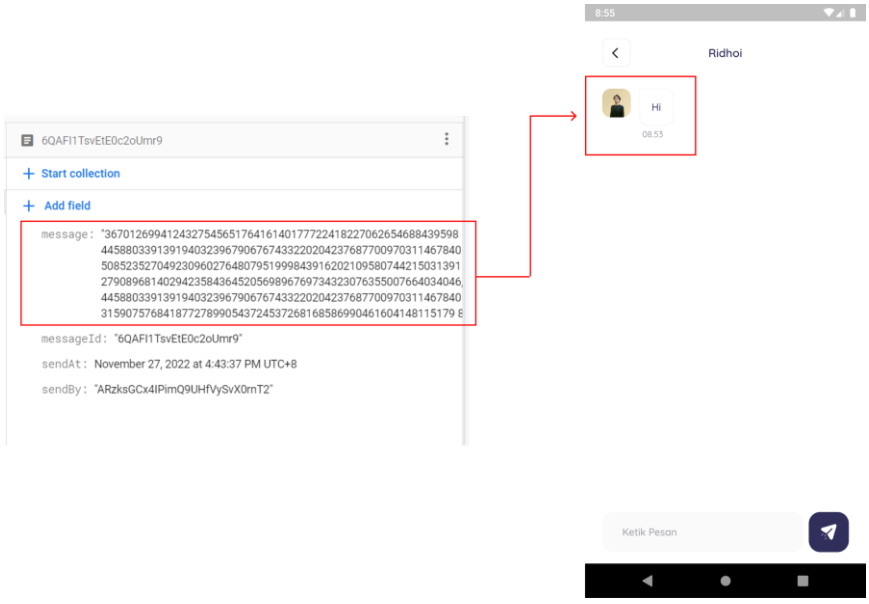
yang dikirim ke server, hasil percobaan pengiriman pesan dapat dilihat pada gambar 25.



Gambar 25 Enkripsi pesan

4.3.3 Dekripsi Pesan

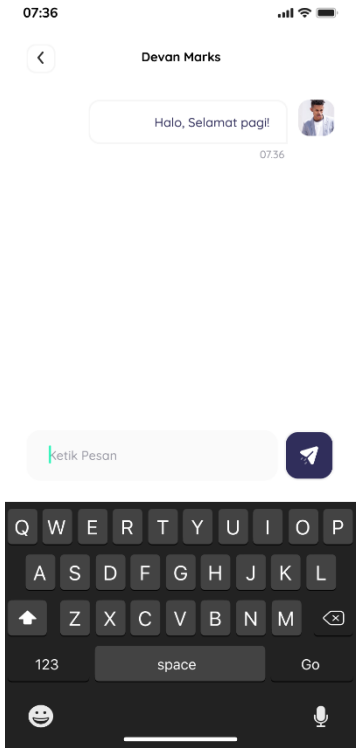
Dekripsi pesan akan dilakukan sistem ketika pengirim pesan mengirimkan pesan ke pengguna dalam bentuk *ciphertext*, kemudian hasil dari dekripsi pesan tersebut akan ditampilkan dilayar pengguna. hasil percobaan penerimaan pesan dapat dilihat pada gambar 26.



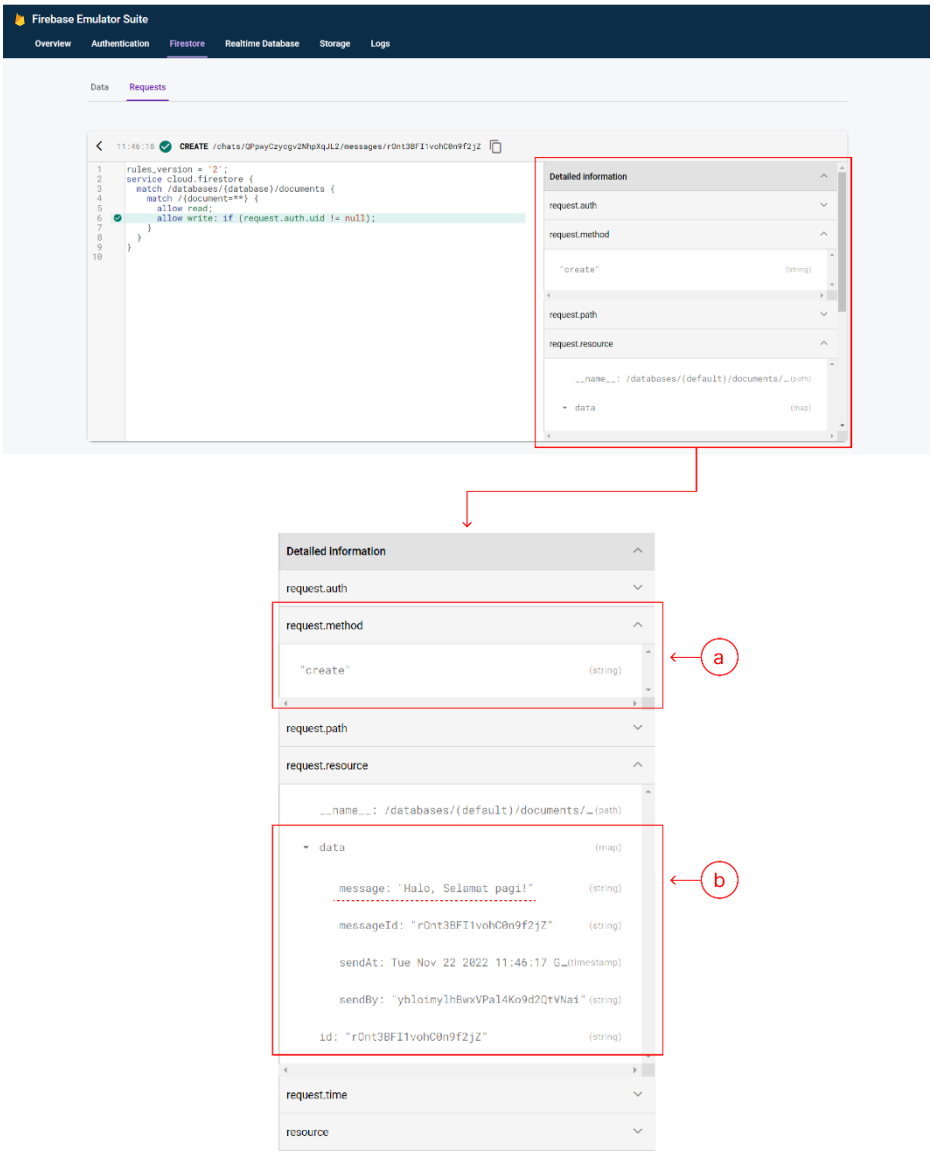
Gambar 26 Dekripsi pesan

4.3.4 *Request Monitor*

Untuk mengetahui apakah pesan yang dikirimkan sudah dalam bentuk *ciphertext* atau tidak, maka pengujian dilakukan dengan memantau *request* yang dilakukan ke *server* menggunakan *Firebase Emulator*. Pada proses pengujian pertama, fungsi enkripsi tidak diaktifkan sehingga pesan yang dikirimkan berupa pesan yang dapat dibaca atau mudah dimengerti. Hasil percobaan pengiriman pesan yang tidak terenkripsi dapat dilihat pada gambar 27 dan 28:



Gambar 27 Pengiriman pesan tanpa enkripsi

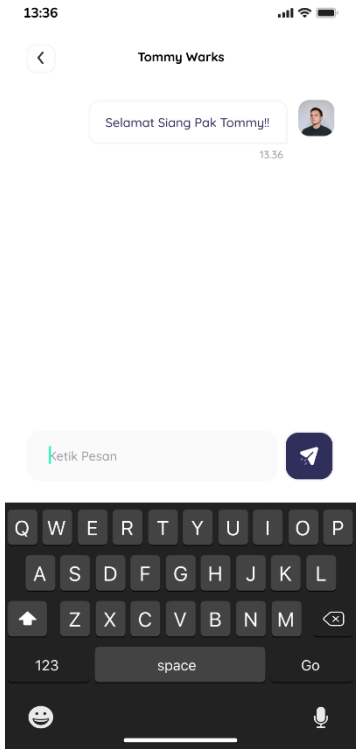


Gambar 28 Hasil *request* pesan tanpa enkripsi

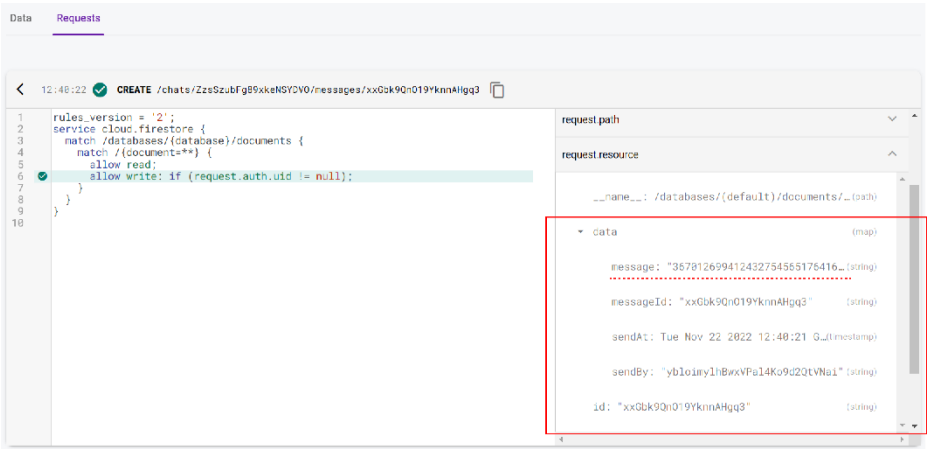
Keterangan dari gambar 28 hasil *request* tanpa enkripsi sebagai berikut:

a. Pada proses pengiriman pesan dilakukan dengan menggunakan *method create*, dimana data yang dikirim ditampung terlebih dahulu pada *database* yang kemudian diambil menggunakan *method get* oleh penerima pesan.
b. *Request* yang dilakukan menampilkan data *message* berisi pesan yang dapat dibaca seperti yang tertera pada gambar 27.

Pengujian selanjutnya dilakukan dengan mengaktifkan fungsi enkripsi. Karena fungsi enkripsi diaktifkan, maka pesan yang dikirim sudah dalam bentuk cipherteks atau tidak dapat dibaca. Hasil percobaan pengiriman pesan yang terenkripsi dapat dilihat pada gambar 29 dan 30.



Gambar 29 Pengiriman pesan terenkripsi



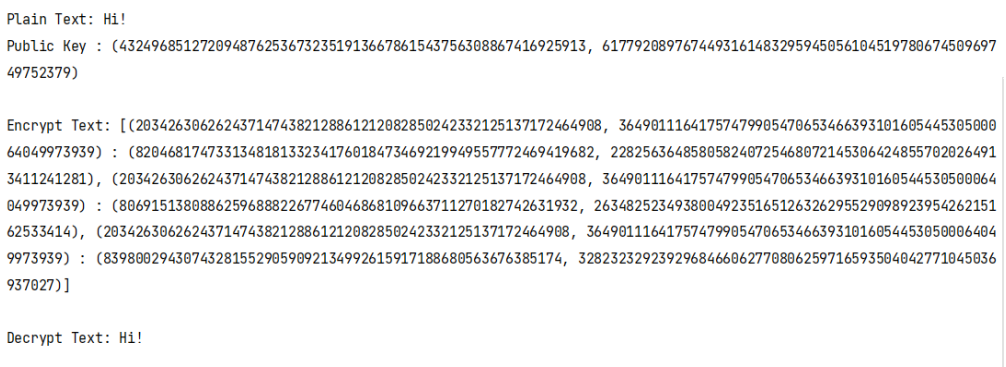
Gambar 30 Hasil *request* pesan terenkripsi

Pada gambar 30 dapat dilihat hasil *request* dengan pesan terenkripsi menampilkan data *message* berisi pesan yang tidak sama dengan pesan yang tertera pada gambar 29.

Berdasarkan hasil pengujian yang telah dilakukan, terdapat perbedaan bentuk pesan hasil *request* ketika fungsi enkripsi tidak diaktifkan dan ketika fungsi enkripsi diaktifkan. Jika fungsi enkripsi tidak diaktifkan, maka pesan dapat dibaca. Sedangkan jika fungsi enkripsi diaktifkan, maka data pesan dari hasil *request* berupa cipherteks sehingga tidak mudah untuk dibaca dan perlu didekripsi terlebih dahulu agar pesan tersebut mudah dibaca.

4.4 Pengujian Enkripsi dan Dekripsi Algoritma *Elliptic Curve Cryptography*

Pengujian enkripsi dan dekripsi pada algoritma *Elliptic Curve Cryptography* (ECC) dengan parameter kurva eliptik Secp192r1, Secp256r1, dan Secp521r1 didapatkan perbandingan hasil *ciphertext* dari proses enkripsi dan kunci publik yang dibangkitkan dan hasil pengujian dengan panjang karakter yang sama dan berbeda sebanyak 15 kali pada waktu proses enkripsi dan dekripsi dari setiap parameter. Salah satu contoh hasil enkripsi, dekripsi dan kunci publik dapat dilihat pada gambar 31.



Gambar 31 Contoh hasil enkripsi, dekripsi dan kunci publik parameter Secp192r1

Pada gambar 31 adalah contoh hasil enkripsi, dekripsi dan kunci publik dengan parameter Secp192r1. Inputan *plaintext* berupa teks yang ditulis langsung dalam program sebagai variabel *string*. Hasil enkripsi dan kunci publik berupa titik yang memiliki koordinat x dan y ditandai dengan pemisah koma (,). Pada hasil enkripsi (*ciphertext*) memiliki 2 pasangan titik dari setiap karakter yang dienkripsi dengan pemisah titik dua (:), titik pertama disebut dengan C1, dan titik kedua disebut dengan C2.

4.4.1 Perbandingan Hasil *Ciphertext* dan Kunci Publik

Perbandingan hasil *ciphertext* dan kunci publik dari proses enkripsi pada parameter kurva eliptik Secp192r1, Secp256r1, dan Secp521r1 dapat dilihat pada tabel 9.

Tabel 9. Perbandingan kunci dengan *ciphertext* hasil enkripsi

Jumlah Karakter	Ukuran Digit *Ciphertext* Tiap Parameter		
	Secp192r1 Ukuran Kunci Publik = 115 digit	Secp256r1 Ukuran Kunci Publik = 153 digit	Secp521r1 Ukuran Kunci Publik = 314 digit
1	232	308	626
10	2318	3075	6269
25	5794	7692	15670
50	11587	15382	31338
75	17380	23081	47004
100	23176	30779	62669
200	46355	61549	125343
300	69539	92330	188027
400	92718	123106	250703
500	115892	153889	313381
600	139068	184677	376055
700	162245	215453	438732

800	185419	246217	501410
900	208597	276987	564089
1000	231781	307764	626771

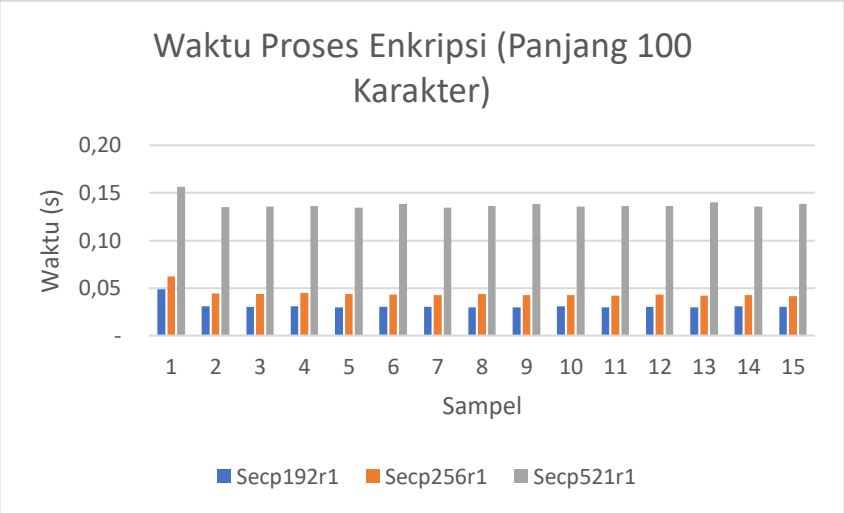
Pada tabel 9 hasil *ciphertext* dari enkripsi dan kunci publik pada setiap parameter menghasilkan digit *ciphertext* dan kunci publik yang berbeda. Pada digit ciphertext dapat memiliki variasi panjang $\pm$ 1-3 digit pada tiap *ciphertext* dari karakter teks yang dienkripsi, sebab pada karakter teks yang dienkripsi menghasilkan digit *ciphertext* yang tidak selalu sama. Dari hasil percobaan ketiga parameter tersebut dapat diketahui perkiraan panjang *ciphertext* dengan cara sebagai berikut:

Panjang hasil *ciphertext*
= Jumlah karakter input $\times$ (Jumlah digit kunci publik $\times$ 2)

4.4.2 Hasil Pengujian Waktu Enkripsi dan Dekripsi

4.4.2.1 Hasil Pengujian dengan Panjang Teks 100 Karakter

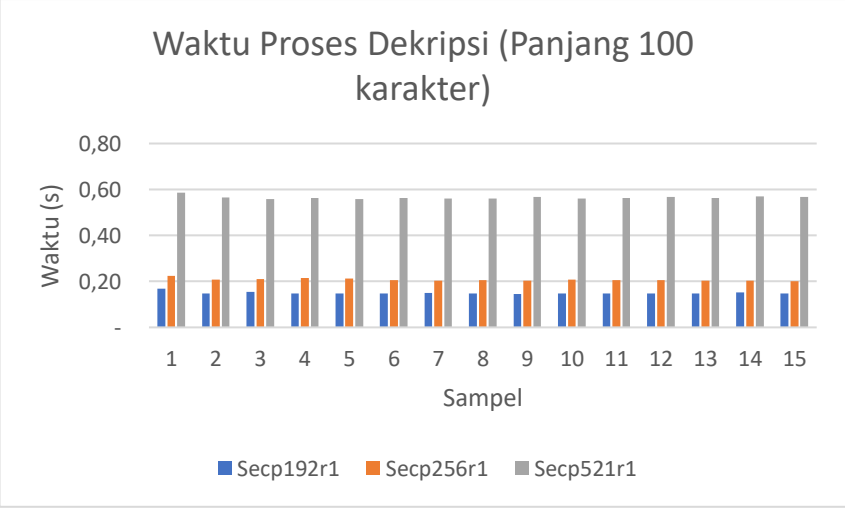
Hasil Pengujian waktu proses enkripsi dan dekripsi dengan panjang teks 100 karakter dilakukan sebanyak 15 kali pada parameter kurva eliptik Secp192r1, Secp256r1, dan Secp521r1 didapatkan hasil perbandingan waktu proses enkripsi dan dekripsi pada gambar 32 dan 33, sedangkan hasil perbandingan rata-rata dan standar deviasi dapat dilihat pada tabel 10 dan 11.



Gambar 32 Perbandingan waktu proses enkripsi dengan panjang 100 karakter

Tabel 10. Perbandingan kecepatan waktu (detik) proses enkripsi dengan panjang 100 karakter

Parameter Kurva Eliptik	Rata-Rata	Standar Deviasi
Spec192r1	0,031431	0,004728
Secp256r1	0,044244	0,005032
Secp521r1	0,137730	0,005385



Gambar 33 Perbandingan waktu proses dekripsi dengan panjang 100 karakter

Tabel 11. Perbandingan kecepatan waktu (detik) proses dekripsi dengan panjang 100 karakter

Parameter Kurva Eliptik	Rata-Rata	Standar Deviasi
Spec192r1	0,149425	0,005633
Secp256r1	0,207345	0,005906
Secp521r1	0,565046	0,006612

Pada tabel 10 dan 11 dapat dilihat bahwa standar deviasi dari ketiga parameter pada proses enkripsi dan dekripsi yang dihasilkan menunjukkan nilai standar deviasi lebih kecil dari pada nilai rata-rata, dimana standar deviasi suatu percobaan semakin rendah, maka data dalam percobaan tersebut semakin mengumpul pada nilai rata-ratanya yang berarti sebaran data bersifat homogen.

Rata-rata dari waktu proses enkripsi dan dekripsi algoritma *Elliptic Curve Cryptography* (ECC) dengan parameter kurva eliptik Secp192r1, Secp256r1, dan Secp521r1 menghasilkan nilai rata-rata yang berbeda. Pada nilai rata-rata enkripsi menunjukkan parameter Secp192r1 lebih rendah dalam artian waktu proses enkripsi lebih cepat dari pada Secp256r1 dengan selisih waktu 0,012813 (28.9% lebih cepat) dan Secp521r1 dengan selisih waktu 0,106299 (77.1% lebih cepat). Selain itu parameter Secp256r1 memiliki selisih waktu proses enkripsi sebesar 0,093486 (67.8% lebih cepat) dibandingkan dengan parameter Secp521r1.

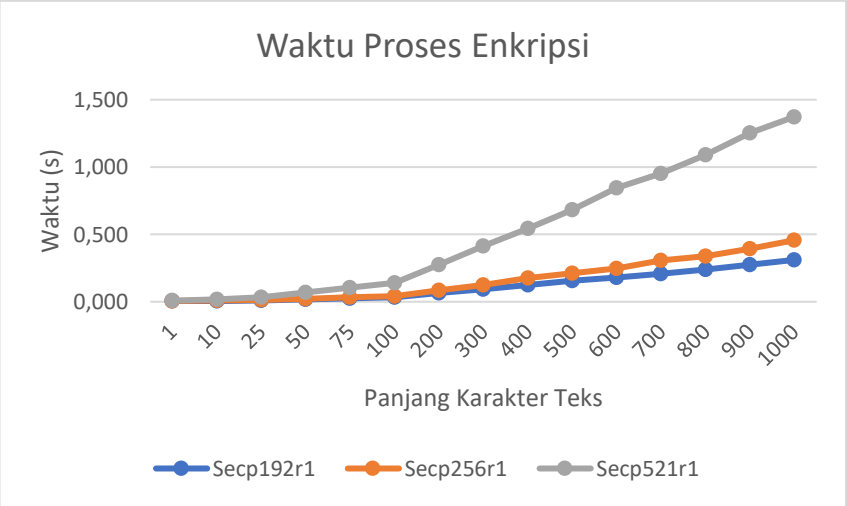
Pada nilai rata-rata waktu dekripsi menunjukkan parameter Secp192r1 lebih rendah dalam artian waktu proses dekripsi lebih cepat dari pada Secp256r1 dengan selisih waktu 0,057920 (27,9% lebih cepat) dan Secp521r1 dengan selisih waktu 0,415621 (73.5% lebih cepat). Selain itu parameter Secp256r1 memiliki selisih waktu proses dekripsi sebesar 0,357701 (63.3% lebih cepat) dibandingkan dengan parameter Secp521r1.

4.4.2.2 Hasil Pengujian dengan Panjang Teks Berbeda

Hasil Pengujian waktu enkripsi dan dekripsi pada parameter kurva eliptik Secp192r1, Secp256r1, dan Secp521r1 dengan panjang karakter teks yang berbeda dapat dilihat pada tabel 12 dan 13.

Tabel 12. Waktu proses enkripsi dengan panjang teks berbeda

Jumlah Karakter	Waktu Proses (detik) Enkripsi		
	Secp192r1	**Secp256r1**	**Secp521r1**
1	0,005522	0,005817	0,008304
10	0,004931	0,010824	0,016495
25	0,010217	0,012874	0,034679
50	0,015928	0,022347	0,068468
75	0,024034	0,031909	0,104731
100	0,032613	0,041437	0,138510
200	0,063477	0,083435	0,274617
300	0,093777	0,125701	0,412002
400	0,124328	0,174322	0,546150
500	0,155400	0,210156	0,683028
600	0,179921	0,247426	0,846712
700	0,209508	0,308625	0,952730
800	0,239424	0,338134	1,092850
900	0,274134	0,393801	1,253591
1000	0,310846	0,457082	1,373762

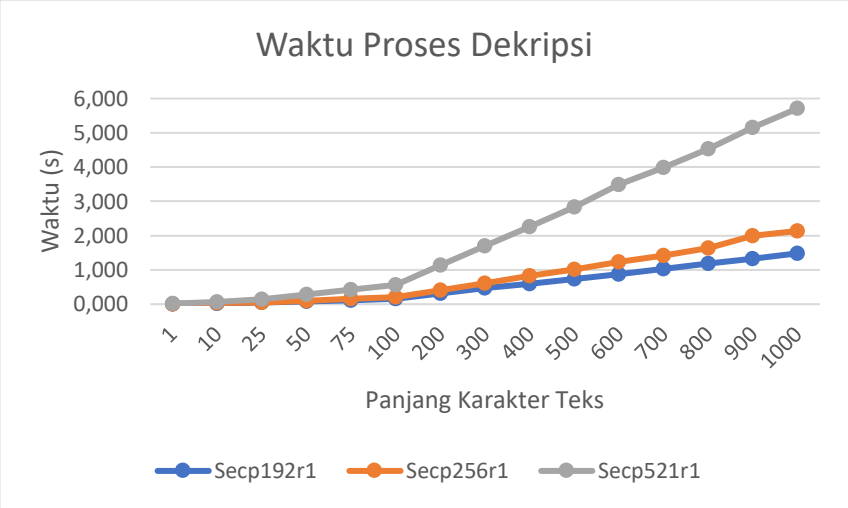


Gambar 34 Perbandingan waktu proses enkripsi dengan panjang karakter teks berbeda

Tabel 13. Waktu proses dekripsi dengan panjang karakter teks berbeda

Jumlah Karakter	Waktu Proses (detik) Dekripsi		
	Secp192r1	**Secp256r1**	**Secp521r1**
1	0,008089	0,009087	0,014383
10	0,020034	0,031451	0,064723
25	0,041962	0,053106	0,146963
50	0,079430	0,102001	0,283162
75	0,117341	0,155020	0,430345
100	0,153620	0,198480	0,566717
200	0,307491	0,406637	1,133566
300	0,462732	0,614453	1,704351
400	0,591328	0,822557	2,261383
500	0,741413	1,014589	2,840887
600	0,880960	1,239297	3,496735

700	1,028992	1,425612	3,993561
800	1,179741	1,631868	4,528521
900	1,332479	1,995076	5,156607
1000	1,483225	2,135863	5,709066



Gambar 35 Perbandingan waktu proses dekripsi dengan panjang karakter teks berbeda

Berdasarkan hasil perbandingan waktu proses enkripsi dan dekripsi dengan panjang karakter teks berbeda pada gambar 34 dan 35, didapatkan parameter Secp192r1 lebih cepat dari parameter Secp256r1 dan Secp521r1 dari setiap pengujian yang dilakukan disebabkan oleh perbedaan ukuran jumlah digit kunci publik yang dihasilkan dan jumlah digit parameter yang digunakan, maka pada proses enkripsi dengan jumlah digit parameter lebih banyak menghasilkan jumlah digit *ciphertext* yang lebih banyak pula, sehingga membutuhkan waktu yang lebih lama. Selain itu pada proses dekripsi dengan jumlah digit kunci publik lebih banyak memerlukan waktu lebih lama dibandingkan jumlah digit yang lebih rendah. Dapat dilihat pada tabel 9 bahwa parameter Secp192r1 menghasilkan *ciphertext* dengan digit paling kecil

disetiap sampel data uji dan kunci publik sebesar 115 digit. Dari penjelasan tersebut dapat disimpulkan bahwa perbedaan waktu proses enkripsi dan dekripsi dari setiap parameter yang berbeda disebabkan oleh besarnya *overhead*. *Overhead* dari proses enkripsi dan dekripsi ditentukan oleh panjang karakter teks dan nilai parameter yang digunakan dalam algoritma *Elliptic Curve Cryptography* (ECC).

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Aplikasi *chat* dengan *end-to-end encryption* telah berhasil dibangun dengan sistem yang memiliki fungsi mengirim dan menerima pesan dengan implementasi teknik kriptografi *Elliptic Curve Cryptography* (ECC). Didalam sistem kriptografi pada aplikasi mencakup proses pembangkitan kunci publik dari masukan kunci privat, proses enkripsi pesan yang dilakukan terlebih dahulu sebelum dikirim ke penerima dan proses dekripsi pesan.

Pengujian keamanan pesan yang dikirim melalui aplikasi *chat* dengan *end-to-end encryption* yang dibangun dilakukan pemantauan *request* ke *server*. Dari hasil pemantauan tersebut dapat diketahui pesan yang dikirim telah berbentuk *ciphertext* dari proses enkripsi atau tidak. Dengan terenkripsinya pesan maka keamanan pesan akan lebih terjaga dari tindakan penyalahgunaan oleh pihak yang tidak berwenang.

Hasil uji kecepatan waktu proses enkripsi dan dekripsi pada algoritma *Elliptic Curve Cryptography* (ECC) dengan panjang teks 100 karakter pada 15 kali percobaan disetiap parameter kurva eliptik yang berbeda menunjukkan parameter Secp192r1 28.9% dan 77.1% lebih cepat dibandingkan dengan Secp256r1 dan Secp521r1 pada proses enkripsi, sedangkan pada proses dekripsi 27,9% dan 73.5% lebih cepat dibandingkan Secp256r1 dan Secp521r1. Perbedaan waktu proses enkripsi dan dekripsi dari setiap parameter yang berbeda disebabkan oleh besarnya *overhead*. *Overhead* dari proses enkripsi dan dekripsi ditentukan oleh panjang karakter pesan dan nilai parameter yang digunakan dalam algoritma *Elliptic Curve Cryptography* (ECC).

5.2 Saran

Dalam penelitian ini masih terdapat banyak kekurangan dan jauh dari kata sempurna, oleh karena itu diperlukan perbaikan dan pengembangan agar sistem

ini dapat menjadi lebih baik lagi. Berikut merupakan beberapa saran untuk penelitian selanjutnya yaitu:

1. Aplikasi diharapkan dapat lebih ditingkatkan lagi sehingga tidak hanya dapat mengenkripsi dan deskripsi pesan teks saja, namun juga dapat mengenkripsi dan deskripsi file, audio, dan video.
2. Aplikasi diharapkan dapat menyediakan fitur voice call dan video call yang terenkripsi agar dapat menunjang komunikasi bagi pengguna aplikasi *chat* ini.
3. Sistem pada aplikasi diharapkan dapat mendukung karakter pesan *unicode* atau lebih dari 127 karakter pada ASCII.
4. Aplikasi dapat mendukung pertukaran pesan untuk lebih dari satu orang atau *multichat*.
5. Diperlukan cara atau metode yang dapat mengurangi ukuran *ciphertext* yang dihasilkan dari proses enkripsi.

DAFTAR PUSTAKA

Adha Bilqis Ibrahim, K., & Gustina, D. (2021). RANCANG BANGUN APLIKASI BERBASIS ANDROID UNTUK BRAND CLOTHING SAND BEACH DENGAN SKEMA DISKON MENGGUNAKAN HUNGARIAN ALGORITHM. *JSI (Jurnal Sistem Informasi) Universitas Suryadarma*, *8*(1), 47–56.

Amrulloh, A., & Ujianto, E. (2019). Kriptografi Simetris Menggunakan Algoritma Vigenere Cipher. *Jurnal CoreIT*, *5*(2). https://program.arfianhidayat.com/kriptografi/vig

Basri. (2016). *KRIPTOGRAFI SIMETRIS DAN ASIMETRIS DALAM PERSPEKTIF KEAMANAN DATA DAN KOMPLEKSITAS KOMPUTASI*. *2*. http://ejournal.fikom-unasman.ac.id

Boruah, D., & Saikia, M. (2014). Implementation of ElGamal Elliptic Curve Cryptography Over Prime Field Using C. *Information Communication and Embedded Systems (ICICES)*.

Br Sembiring, M. (2015). *Elliptic Curve Cryptography (Ecc) Pada Proses Pertukaran Kunci Publik Diffie-Hellman* (Vol. 1).

Damanik, P. P. (2019). Implementasi Algoritma Elliptic Curve Cryptography (ECC) Untuk Penyandian Pesan Pada Aplikasi Chatting Client Server Berbasis Desktop. *JURIKOM (Jurnal Riset Komputer)*, *6*(4), 395–400.

Diro, A. A., Chilamkurti, N., & Veeraraghavan, P. (2017). Elliptic curve based cybersecurity schemes for publish-subscribe internet of things. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, *199*, 258–268. https://doi.org/10.1007/978-3-319-60717-7_26

Fahlevie, F. H. (2012). *PENGEMBANGAN APLIKASI CHATTING BERBASIS KOMUNITAS MENGGUNAKAN METODE SOCKET DAN WEIGHTED SUM MODEL STUDI KASUS UIN MALIKI MALANG*. Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Fatonah, Iskandar Mulyana, D., Heryani, A. P., & Khoirunnisa, V. (2022). Implementasi Metode Rivest Shamir Adleman untuk Enkripsi dan Dekripsi Text. *Jurnal Informatika Dan Teknologi Komputer*, *3*(1), 32–39. https://ejurnalunsam.id/index.php/jicom/

Hammi, B., Fayad, A., Khatoun, R., Zeadally, S., & Begriche, Y. (2020). A Lightweight ECC-Based Authentication Scheme for Internet of Things (IoT). *IEEE Systems Journal*, *14*(3), 3440–3450. https://doi.org/10.1109/JSYST.2020.2970167

Kadi, D. (2017). *Pengembangan Aplikasi Mobile Objek Wisata Secara Real Time Dengan Augmented Reality di Kabupaten Sumba Barat Daya*. UNIVERSITAS ATMA JAYA YOGYAKARTA.

Luthfi, K. (2020). Rancang Bangun Aplikasi Sistem Transaksi Laundry Berbasis Mobile Menggunakan Flutter. *Jurnal Manajemen Informatika*.

Mardianto, E., Uzzin, I., & Setiowati, Y. (2015). *Enkripsi SMS menggunakan ECC*.

Nugroho, A. D., & Munir, R. (2015). *Aplikasi Enkripsi Instant Messaging Pada Perangkat Mobile Dengan Menggunakan Algoritma Elliptic Curve Cryptography (ECC)*.

Panggabean, A. R. (2020). Implementasi Algoritma Paillier Cryptosystem Untuk Keamanan Data Video Mpeg Pada Aplikasi Chat. *Jurnal Informasi Dan Teknologi Ilmiah (INTI)*, *8*(1), 1–6.

Perdana, D., Purwiko, P., Dewanta, F., & Afianti, F. (2022). Analysis of Using ECC in Authentication Systems in IoT. *JURNAL MULTIMEDIA NETWORKING INFORMATICS*, *8*, 42–49.

Pratiwi, A. J., & Asmunin. (2022). Penggunaan QR code Berbasis Kriptografi Menggunakan Algoritma Elliptic Curve Criptography. *Journal of Informatics and Computer Science*, *03*(4).

Priyantono, M. G. (2019). *Pembangunan Aplikasi Perencanaan Keuangan Pribadi Menggunakan Teknologi Firebase Cloud Messaging Dan Api Toko Online Berbasis Android*. Universitas Komputer Indonesia.

Qazi, R., Qureshi, K. N., Bashir, F., Islam, N. U., Iqbal, S., & Arshad, A. (2021). Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 547–566. https://doi.org/10.1007/s12652-020-02020-z

Randi, A., Lazuardy, K., Chandra, S., & Dharma, A. (2020). Implementasi Algoritma Advanced Encryption Standard pada Aplikasi Chatting berbasis Android. *JIKOMSI Jurnal Ilmu Komputer Dan Sistem Informasi*, *3*(2), 1–10.

Santoso, H., & Siambaton, M. Z. (2020). APLIKASI PENGAMANAN EKSTENSI FILE MENGGUNAKAN KRIPTOGRAFI ONE TIME PAD (OTP) DAN ELLIPTIC CURVE CRYPTOGRAPHY (ECC). *JISTech (Journal of Islamic Science and Technology)*, *5*(1), 22–38. http://jurnal.uinsu.ac.id/index.php/jistech

Santria, U., & Arsoetar, N. (2017). *Penggunaan Enkripsi End-to-End dalam Pengamanan Pesan dan Video Call pada Whatsapp*.

Sumandri, S. (2017). Studi Model Algoritma Kriptografi Klasik dan Modern. Semin. Mat. dan Pendidik. Mat. UNY, 265-272.

LAMPIRAN

Lampiran 1 *Source Code*

Source code penelitian tersedia untuk publik pada tautan berikut: https://github.com/rrdhoi/Elliptic-Curve-Cryptography.git