IMPLEMENTASI RSA CRYPTOGRAPHY ALGORITMA DALAM ENKRIPSI DATA UNTUK MANIPULASI LOKASI BERDASARKAN ALAMAT IP

SKRIPSI

DISUSUN OLEH

MUHAMMAD BINTANG HARAHAP

NPM. 209020108

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2024

IMPLEMENTASI RSA CRYPTOGRAPHY ALGORITMA DALAM ENKRIPSI DATA UNTUK MANIPULASI LOKASI BERDASARKAN ALAMAT IP

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

MUHAMMAD BINTANG HARAHAP
NPM. ISI 2009020108

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2024

LEMBAR PENGESAHAN

Judul Skripsi	:	IMPLEMENTASI RSA CRYPTOGRAPHY ALGORITMA DALAM ENKRIPSI DATA UNTUK MANIPULASI LOKASI BERDASARKAN ALAMAT IP
Nama Mahasiswa	:	MUHAMMAD BINTANG HARAHAP
NPM	:	209020108
Program Studi	:	TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing

(Amrullah, S.Kom., M.Kom)
NIDN. 0125118604

Ketua Program Studi

(Fatma Sari Hutagalung, S.Kom, M.Kom)
NIDN. 0117019301

Dekan

(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

IMPLEMENTASI RSA CRYPTOGRAPHY ALGORITMA DALAM ENKRIPSI DATA UNTUK MANIPULASI LOKASI BERDASARKAN ALAMAT IP

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 6 Juli 2024

Yang membuat pernyataan

Muhammad Bintang Harahap

NPM. 2009020108

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Muhammad Bintang Harahap
NPM	: 2009020108
Program Studi	: Teknologi Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

IMPLEMENTASI RSA CRYPTOGRAPHY ALGORITMA DALAM ENKRIPSI DATA UNTUK MANIPULASI LOKASI BERDASARKAN ALAMAT IP

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 6 Juli 2024

Yang membuat pernyataan

Muhammad Bintang Harahap

NPM. 2009020108

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Muhammad Bintang Harahap
Tempat dan Tanggal Lahir	: Bandar Setia, 21-12-2001
Alamat Rumah	: Jln.M.Yacub Lubis Gg.Berkah No.257
Telepon/Faks/HP	: 082277494274
E-mail	: muhammadbintang.2112@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SDN 101765	TAMAT: 21 Juni 2014
SMP	: SMP PRAYATNA MEDAN	TAMAT: 2 Juni 2017
SMA	: SMAN 2 PERCUT SEI TUAN	TAMAT: 2 Mei 2020

KATA PENGANTAR

Pendahuluan

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom, M.Kom Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Basri, S.Si, M.Kom Sekretaris Program Studi Teknologi Infomasi
5. Bapak Amrullah, S.Kom, M.Kom Selaku Dosen Pembimbing saya yang telah mendidik dan memberikan arahan.
6. Kepada kedua orang tua saya bapak Jaharuddin Hrp dan ibu Nisma Nst, yang telah mendukung sepenuh hati dan sabar menunggu saya wisuda
7. Terimakasih kepada Widya Silviani yang telah memberikan semangat yang luar biasa
8. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini

IMPLEMENTASI RSA CRYPTOGRAPHY ALGORITMA DALAM ENKRIPSI DATA UNTUK MANIPULASI LOKASI BERDASARKAN ALAMAT IP

ABSTRAK

Dalam era digital ini, keamanan data merupakan hal yang sangat penting dalam era digital, sehingga dimana informasi pribadi dan sensitif sering kali dikirim dan disimpan secara elektronik. Penelitian ini berfokus pada peningkatan keamanan data geolokasi yang diperoleh dari alamat IP melalui penggunaan algoritma enkripsi RSA. Salah satu metode yang efektif untuk melindungi data adalah melalui sebuah enkripsi, yang mengubah data asli menjadi bentuk yang tidak dapat dibaca tanpa kunci enkripsi yang tepat. Salah satu mekanisme untuk meningkatkan keamanan data adalah dengan menggunakan teknologi enkripsi. Data-data yang disimpan dalam database diubah sedemikian rupa sehingga tidak mudah dibaca. Jadi enkripsi adalah proses yang dilakukan untuk mengamankan sebuah data (yang disebut plaintext) menjadi data yang tersembunyi (disebut ciphertext). Ciphertext adalah data yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah “encipher”. Proses sebaliknya, untuk mengubah ciphertext menjadi plaintext, disebut dekripsi (decryption). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah “decipher”algoritma RSA ini sistem akan mengenkripsi data asli yang diinputkan peneliti menjadi chiperteks dengan menggunakan key, kemudian mengirimkan kepada orang lain ataupun rekannya. Untuk penerimaan data asli dideskripsi menjadi plainteks menggunakan key juga oleh penerima sehingga pengiriman informasi atau pemanfaatan informasi melalui keamanan algoritma RSA menjadi lebih muda dipahami oleh penerima ataupun pengguna tersebutHasil penelitian dapat membantu memperluas pengetahuan peneliti mengenal implementasi RSA Cryptography dalam enkripsi data untuk manipulasi lokasi berdasarkan alamat IP.Dalam peneltian ini dapat memberikan kesimpulan bahwa penggunaan algoritma RSA efektif dalam melindungi data geolokasi dari akses yang tidak sah. Proses enkripsi dan deskripsi berjalan baik, memastikan bahwa data hanya dapat diakses oleh pihak yang memiliki kunci deskripsi yang sesuai. Penguji keamanan menunjukkan bahwa enkripsi RSA dengan panjang kunci 2048-bit dapat memberikan perlindungan data yang baik terhadap serangan brute force. Selain itu juga penelitian ini menunjukkan bahwa data geolokasi yang terenkripsi dapat disimpan dengan aman didalam basis data, yang dapat menjaga kerahasian dan integritas informasi, dan memastikan bahwa hanya pengguna yang sah yang dapat mengakses dan megelola data terenkripsi.

Kata Kunci: Keamanan Data;Geolokasi;Alamat IP;Algoritma RSA

IMPLEMENTATION OF RSA CRYPTOGRAPHY ALGORITHM IN DATA ENCRYPTION FOR MANIPULATION LOCATION BASED ON IP ADDRESS

ABSTRACT

In this digital age, data security is of paramount importance, where personal and sensitive information is often sent and stored electronically. This research focuses on improving the security of geolocation data obtained from IP addresses through the use of RSA encryption algorithm. One effective method of protecting data is through encryption, which converts the original data into an unreadable form without the right encryption key. One mechanism to improve data security is to use encryption technology. The data stored in the database is altered in such a way that it cannot be easily read. So encryption is a process that is done to secure data (called plaintext) into hidden data (called ciphertext). Ciphertext is data that cannot be read easily. According to ISO 7498-2, the more appropriate terminology to use is “encipher”. The reverse process, to convert ciphertext into plaintext, is called decryption. According to ISO 7498-2, the more appropriate terminology for this process is “decipher”. This RSA algorithm system will encrypt the original data inputted by the researcher into chipertext using the key, then send it to other people or colleagues. For the reception of the original data, it is decrypted into plaintext using the key also by the recipient so that the transmission of information or the utilization of information through the security of the RSA algorithm becomes younger understood by the recipient or user.The results of the study can help expand the knowledge of researchers about the implementation of RSA Cryptography in data encryption for location manipulation based on IP addresses.In this research, it can provide a conclusion that the use of the RSA algorithm is effective in protecting geolocation data from unauthorized access. The encryption and description process works well, ensuring that the data can only be accessed by those who have the appropriate description key. Security testers showed that RSA encryption with 2048-bit key length can provide good data protection against brute force attacks. In addition, this research shows that the encrypted geolocation data can be securely stored in the database, which can maintain the confidentiality and integrity of the information, and ensure that only authorized users can access and manage the encrypted data.

Keywords: Data Security;Geolocation;IP Address;RSA Algorithm

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam era digital ini, keamanan data merupakan hal yang sangat penting dalam era digital, sehingga dimana informasi pribadi dan sensitif sering kali dikirim dan disimpan secara elektronik. Penelitian ini berfokus pada peningkatan keamanan data geolokasi yang diperoleh dari alamat IP melalui penggunaan algoritma enkripsi RSA. Salah satu metode yang efektif untuk melindungi data adalah melalui sebuah enkripsi, yang mengubah data asli menjadi bentuk yang tidak dapat dibaca tanpa kunci enkripsi yang tepat. Salah satu mekanisme untuk meningkatkan keamanan data adalah dengan menggunakan teknologi enkripsi. Data-data yang disimpan dalam database diubah sedemikian rupa sehingga tidak mudah dibaca. Jadi enkripsi adalah proses yang dilakukan untuk mengamankan sebuah data (yang disebut plaintext) menjadi data yang tersembunyi (disebut ciphertext). Ciphertext adalah data yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah “encipher”. Proses sebaliknya, untuk mengubah ciphertext menjadi plaintext, disebut dekripsi (decryption). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah “decipher”. (Rumetna, 2021)

Pengetahuan yang mempelajari tentang enkripsi adalah kriptografi. Yang dimaksud dengan kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Akan tetapi, tidak semua aspek keamanan informasi bisa diatasi dengan kriptografi.

Kriptografi merupakan suatu ilmu yang mempelajari teknik dalam pengenkripsian data acak yang yang diubah kedalam sesuatu bacaan yang sulit dipahami atau sulit dibaca menggunakan suatu kunci, dimana bacaan yang sulit dipahami dan sulit dibaca tidak akan terbaca oleh user yang tidak memiliki kunci dekripsi. (Chafid & Soffiana, 2022)

Dalam peneletian ini, alamat IP (Internet Protocol) menjadi komponen yang dibutuhkan untuk memungkinkan komunikasi dan identifikasi perangkat di jaringan internet. Alamat IP tidak hanya berfungsi sebagai pengenal untuk sebuah perangkat, tetapi juga dapat digunakan untuk menentukan lokasi geografis perangkat tersebut melalui teknik geolokasi berbasis IP. Dengan adanya manfaat yang ditawarkan, muncul pula sejumlah masalah terkait privasi dan keamanan data lokasi. Informasi geolokasi yang diperoleh dari alamat IP dapat dieksploitasi oleh pihak tidak bertanggung jawab untuk tujuan yang merugikan, seperti pelacakan tanpa izin, pencurian identitas, dan serangan siber lainnya. Oleh karena itu, perlindungan data lokasi menjadi sangat penting untuk menghindari penyalahgunaan informasi yang dapat merugikan individu maupun organisasi. (Studi et al., 2021)

Salah satu solusi yang dibutuhkan adalah dengan menerapkan algoritma RSA, yang dimana RSA (Rivest Shamir Adleman) merupakan salah satu algoritma public key yang populer dipakai dan bahkan hingga saat ini Algoritma RSA masih dianggap aman adalah perluasan dari caesar cipher, yang menggalikan plainteks dengan sebuah nilai dan menambahkannya dengan sebuah pergeseran. Kecurangan data tersebut dapat diatasi dengan memanfaatkan metode RSA secara enkrip dan deskrip. Oleh karena itu dibutuhkan suatu sistem keamanan yang

mampu menjaga kerahasiaan data dari ancaman lain yang dilakukan oleh pihak yang tidak bertanggung jawab. Dengan memanfaatkan algoritma RSA ini sistem akan mengenkripsi data asli yang diinputkan peneliti menjadi chiperteks dengan menggunakan key, kemudian mengirimkan kepada orang lain ataupun rekannya. Untuk penerimaan data asli dideskripsi menjadi plainteks menggunakan key juga oleh penerima sehingga pengiriman informasi atau pemanfaatan informasi melalui keamanan algoritma RSA menjadi lebih muda dipahami oleh penerima ataupun pengguna tersebut. (Ulfah Indriani, Ommi Alfina, 2021)

Dari beberapa permasalahan yang dipaparkan diatas mana peneliti tertarik untuk melakukan penelitian dengan judul "Implementasi RSA CRYPTOGRAPHY ALGORITMA Dalam Enkripsi Data Untuk Manipulasi Lokasi Berdasarkan Alamat IP "

1.2. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang masalah diatas, maka rumusan masalah yang dapat dikaji pada penelitian ini adalah bagaimana peneliti untuk dapat mengimplementasikan algoritma RSA untuk mengenkripsi data geolokasi yang diperoleh dari alamat IP untuk dapat meningkatkan keamanan dan privasi data tersebut.

1.3. Batasan Masalah

Berdasarkan Batasan Masalah dalam Penelitian ini adalah sebagai berikut :

1. Peneliti hanya fokus pada implementasi algoritma RSA untuk enkripsi data geolokasi berdasarkan dari alamat IP.
2. Data yang akan dienkripsi terbatas pada informasi geolokasi yang diperoleh melalui teknik geolokasi berbasis alamat IP.
3. Penelitian ini tidak mencakup aspek keamanan jaringan secara keseluruhan, melainkan hanya fokus pada enkripsi data geolokasi berdasarkan alamat IP.
4. Menggunakan Bahasa Pemograman Python

1.4. Tujuan Penelitian

Berdasarkan Rumusan Masalah diatas, adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Untuk Mengetahui Bagaimana Implementasi Algoritma RSA Dapat Mengenkripsi Data Geolokasi Dengan Baik Berdasarkan IP ?
2. Untuk Mengetahui Bagaimana Implementasi Algoritma RSA Dapat Memastikan Bahwa Data Geolokasi Terenkripsi Dengan Baik, Sehingga

Hanya Dapat Diakses Oleh Pihak yang Memiliki Kunci Deskripsi yang Sah ?

3. Untuk Mengetahui Bagaimana Penerapan Algoritma RSA pada Data Geolokasi Berdasarkan Alamat IP Dengan Menggunakan Python?

1.5. Manfaat Penelitian

Hasil studi ini dinantikan dapat memperkaya ilmu pengetahuan dan bisa digunakan sebagai rujukan atas kajian yang berkaitan dengan Implementasi RSA Cryptography Algoritma Dalam Enkripsi Data Untuk Manipulasi Lokasi Berdasarkan Alamat IP.

1. Manfaat Praktis

a. Bagi Peneliti

Hasil penelitian dapat membantu memperluas pengetahuan peneliti mengenal implementasi RSA Cryptography dalam enkripsi data untuk manipulasi lokasi berdasarkan alamat IP.

b. Bagi Universitas

Hasil penelitian ini dapat menjadi referensi tambahan diperpustakaan dan perbandingan dalam penelitian-penelitian berikutnya.

c. Bagi Pengguna Internet

Hasil penelitian ini dapat meningkatkan pemahaman tentang bagaimana cara mengenkripsi data dengan menggunakan algoritma RSA guna untuk menjaga keamanan dan privasi data.

BAB II

LANDASAN TEORI

2.1. Enkripsi

Enkrip simerupakan proses yang sangat penting dalam kriptografi supaya keamanan data yang dikirimkan bisa terjaga kerahasian-nya. Pesan asli (plaintext)diubah menjadi kode-kode yang tidak dimengerti. Enkripsibisa diartikan dengan chipper atau kode. Sama halnya dengan manusia yang tidak mengerti sebuah kata, manusia akan dapat melihatnya di dalam kamus atau daftar istilah-istilah.Enkripsi merupakan sebuah proses untuk menyembunyikan pesan atau sebuah teks asli dengan cara membuat pesan atau teks tersebut terlihat acak dan sulit untuk dibaca. (Ilmiah & Indonesia, 2023)

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti. Enkripsi dapat diartikan sebagai kode atau cipher. Enkripsi merupakan sebuah system dengan pengkodean menggunakan suatu tabel atau kamus yang telah didefinisikan untuk mengganti kata dari informasi yang dikirim. Sebuah cipher menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (stream) bit dari sebuah pesan menjadi cryptogram yang tidak dimengerti (unintelligible). Enkripsi dimaksudkan untuk melindungi informasi agar tidak terlihat oleh orang atau pihak yang tidak berhak. (TRIANA, 2020)

Enkripsi berfungsi untuk menyandikan data ataupun informasi maka orang yangtidak berkepentingan tidak berhak membacanya. Maka, dengan adanya data anda dienkripsi dengan kunci. Untuk membuka data atau pesan teks, kuncinya sama dengan kunci yang digunakan untuk enkripsi (untuk enkripsi kunci privat)

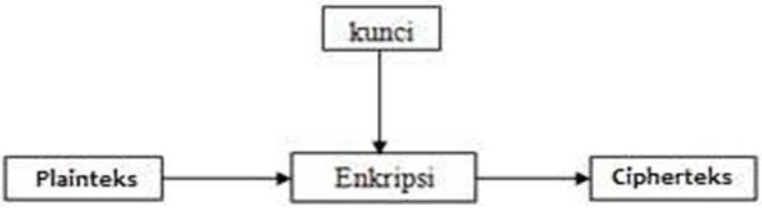
atau memiliki kunci yang berbeda (untuk enkripsi kunci publik). (Feraldi et al., 2021)

2.2. Deskripsi

Menurut Ziliwu dkk (2022)deskripsi merupakan kegiatan yang bertujuan untuk mengembalikan pesan yang telah tersandi atau terenkripsi menjadi pesan asli atau plaintext. Proses mengembalikan isi pesan tersandi menggunakan kode yang telah ditentukan sebelumnya. Dekripsi merupakan kebalikan dari enkripsi, yang merupakan proses mengubah pesan asli menjadi pesan tersandi atau ciphertext. Proses mengubah plaintext menjadi ciphertext disebut enkripsi, sedangkan proses mengubah ciphertext menjadi plaintext disebut dekripsi. Beberapa perusahaan juga mengenkripsi data untuk perlindungan umum data perusahaan dan rahasia dagang. Jika data ini perlu dapat dilihat, mungkin memerlukan dekripsi. Jika kode sandi atau kunci dekripsi tidak tersedia, perangkat lunak khusus mungkin diperlukan untuk mendekripsi data menggunakan algoritme untuk memecahkan dekripsi dan membuat data dapat dibaca. (Harun Alfirdaus et al., 2023)

Dekripsi adalah proses mengubah data yang telah dibuat tidak dapat dibaca melalui enkripsi kembali ke bentuk yang tidak dienkripsi. Dalam dekripsi, sistem mengekstrak dan mengonversi data yang rusak dan mengubahnya menjadi teks dan gambar yang mudah dimengerti tidak hanya oleh pembaca tetapi juga oleh sistem. Dekripsi dapat dilakukan secara manual atau otomatis. Ini juga dapat dilakukan dengan seperangkat kunci atau kata sandi. Salah satu alasan utama untuk menerapkan sistem enkripsi-dekripsi adalah privasi. Ketika informasi menyebar melalui World Wide Web, informasi tersebut menjadi subyek

pengawasan dan akses dari individu atau organisasi yang tidak berwenang. Akibatnya, data dienkripsi untuk mengurangi kehilangan dan pencurian data. Beberapa item umum yang dienkripsi termasuk pesan email, file teks, gambar, data pengguna, dan direktori. Orang yang bertanggung jawab atas dekripsi menerima prompt atau jendela di mana kata sandi dapat dimasukkan untuk mengakses informasi yang dienkripsi.



Gambar 2.2 Proses Enkripsi dan Deskripsi

2.3. Rivest Shamir Adleman (RSA)

RSA merupakan salah satu dari Public Key Cryptosystem yang sangat sering digunakan untuk memberikan kerahasiaan terhadap keaslian suatu data digital. Keamanan enkripsi dan dekripsi data model ini terletak pada kesulitan untuk memfaktorkan modulus n yang sangat besar. Dalam kriptografi, RSA adalah algoritma untuk enkripsi kunci publik. Algoritma ini adalah algoritma pertama yang diketahui paling cocok untuk menandai (signing) dan untuk enkripsi dan salah satu penemuan besar pertaman dalam kriptografi kunci publik. RSA masih digunakan secara luas dalam protokol-protokol perdagangan elektronik dan dipercayai sangat aman karena diberikan kunci-kunci yang cukup panjang dan penerapan-penerapannya yang sangat mutakhir. (Rezki & Siahaan, 2021)

Gambar 2.3 Contoh Kunci Privat dan Publik

Dari sekian banyak algoritma kriptografi kunci publik yang pernah dibuat, algoritma yang paling populer adalah algoritma RSA. Algoritma ini melakukan pemfaktoran bilangan yang sangat besar. Oleh karena alasan tersebut RSA dianggap aman. Untuk membangkitkan dua kunci, dipilih dua bilangan prima acak yang besar. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (Massachussets Institute of Technology) pada tahun 1976, yaitu : Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. Kekuatan algoritma RSA terletak pada sulitnya memfaktorkan bilangan besar menjadi faktor-faktor bilangan primanya, sehingga semakin besar bilangan prima yang digunakan semakin baik atau aman. Dalam kriptografi menggunakan algoritma RSA terdapat tiga proses yaitu proses pembangkitan kunci publik dan kunci privat, proses enkripsi, dan proses dekripsi. (Andika, 2021)

2.4. Cara Kerja Algoritma Rivest Shamir Adleman (RSA)

Algoritma RSA (Rivest-Shamir-Adleman) adalah algoritma kriptografi kunci asimetris yang digunakan untuk enkripsi dan dekripsi pesan. Algoritma ini menggunakan kunci publik dan kunci privat, dimana kunci publik digunakan untuk enkripsi dan kunci privat digunakan untuk dekripsi. Cara kerja algoritma RSA meliputi beberapa tahapan, yaitu:

1. Ekspansi kunci: Memilih dua bilangan prima berukuran besar p dan q, dan hitung n = p x q. Hitung juga m = (p-1) x (q-1).
2. Memilih e: Memilih e yang relatif prima terhadap m, dan e harus tidak sama dengan 1.
3. Memilih d: Memilih d yang relatif prima terhadap m, dan d harus tidak sama dengan 1.
4. Enkripsi: Menghitung kunci publik (e, n), dan menggunakan kunci publik untuk menghasilkan pesan enkripsi (c).
5. Dekripsi: Menghitung kunci privat (d, n), dan menggunakan kunci privat untuk menghasilkan pesan dekripsi (m).

Algoritma RSA digunakan secara luas dalam keamanan data digital, seperti protokol electronic commerce, tanda tangan digital, autentikasi TV berbayar, protokol SSL, dan keamanan kartu elektronik. Keamanan algoritma RSA terletak pada proses eksponensial, dan pemfaktoran bilangan menjadi 2 bilangan prima yang hingga kini perlu waktu yang lama untuk melakukan pemfaktorannya. (Rezki & Siahaan, 2021)

2.5. Cryptography

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namunpada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematikauntuk berurusan dengankeamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas(Nurhasan, 2019). Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakankeamanan informasi. Ilmu

cryptography juga adalah suatu teknik untuk mengamankan data atau pesan dengan melakukan proses enkripsi dan proses deskripsi. (Fauzi, 2023)

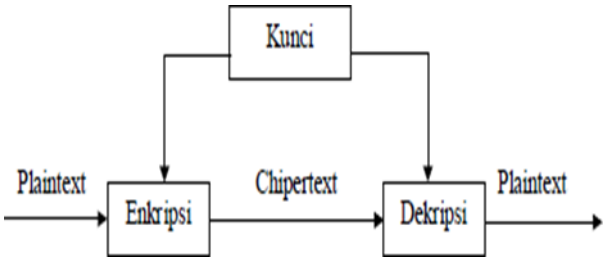
Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti. Enkripsi dapat diartikan sebagai kode atau cipher. Enkripsi merupakan sebuah system dengan pengkodean menggunakan suatu tabel atau kamus yang telah didefinisikan untuk mengganti kata dari informasi yang dikirim. Sebuah cipher menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (stream) bit dari sebuah pesan menjadi cryptogram yang tidak dimengerti (unintelligible). Enkripsi dimaksudkan untuk melindungi informasi agar tidak terlihat oleh orang atau pihak yang tidak berhak.

Dekripsi adalah proses dengan algoritma yang sama untuk mengembalikkan informasi teracak menjadi bentuk aslinya. Algoritma yang digunakan harus terdiri dari susunan prosedur yang direncanakan secara hati-hati yang harus secara efektif menghasilkan sebuah bentuk terenkripsi yang tidak bisa dikembalikan, bahkan sekalipun dengan algoritma yang sama.

Algoritma cryptography bukanlah ditentukan oleh kerumitan dalam mengelola data atau pesan yang akan disampaikan. (TRIANA, 2020) Tetapi, algoritma tersebut memiliki 4 persyaratan sebagai berikut :

1. Kerahasiaan. Pesan (plaintext) hanya bisa dibaca antara dua pihak yang memiliki kewenangan.
2. Autentifikasi. Pengirim pesan harus dapat diidentifikasi dengan pasti dan penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.

3. Integritas. Penerima pesan harus bisa memastikan bahwa pesan yang dia terima bukanlah dimodifikasi ketika saat melakukan proses transmisi data.
4. Non-Repudiation. Pengirim pesan harus tidak bisa menyangkal pesan yang telah dikirim.



Gambar 2.5 Proses Kriptografi

2.6. Jenis-Jenis Serangan Cryptography

Walaupun telah dilakukannya pengaman data dengan teknik crypotography, namun masih terdapat beberapa kemungkinan serangan yang dapat dilakukan terhadap pesan yang sudah di enkripsi berdasarkan ketersediaan data yang ada dan tingkat kesulitannya bagi penyerang, (Waruwu & Hondro, 2024) adalah :

1. Ciphertext only attack, penyerang hanya mendapatkan ciphertext dari sejumlah pesan yang seluruhnya telah dienkripsi menggunakan algoritma yang sama. Sehingga, metode yang digunakan untuk memecahkannya adalah exhaustive key search, yaitu mencoba semua kemungkinan yang ada untuk menemukan kunci.
2. Known plaintext attack, dimana penyerang selain mendapatkan sandi, juga mendapatkan pesan asli. Terkadang disebut pula clear-text attack
3. Choosen plaintext attack, sama dengan known plaintext attack, namun penyerang bahkan dapat memilih penggalan mana dari pesan asli yang akan disandikan. Serangan jenis ini lebih hebat dari pada known-plaintext

attack, karena kriptoanalis dapat memilih plaintext tertentu untuk dienkripsikan, yaitu plaintext-plaintext yang lebih mengarahkan penemuan kunci.

4. Chosen- ciphertext attack. Pada tipe ini, kriptoanalis dapat memilih ciphertext yang berbeda untuk didekripsi dan memiliki akses atas plaintext yang didekripsi.
5. Chosen-key attack. Kriptoanalis pada tipe penyerangan ini memiliki pengetahuan tentang hubungan antara kunci-kunci yang berbeda dan memilih kunci yang tepat untuk mendekripsi pesan.
6. Rubber-hose cryptanalysis. Pada tipe penyerangan ini, kriptoanalis mengancam, menyiksa, memeras, memaksa, atau bahkan menyogok seseorang hingga mereka memberikan kuncinya. Ini adalah cara yang paling ampuh untuk mendapatkan kunci.
7. Adaptive – chosen – plaintext attack. Penyerangan tipe ini merupakan suatu kasus khusus chosen-plaintext attack. Kriptoanalis tidak hanya dapat memilih plaintext yang dienkripsi, ia pun memiliki kemampuan untuk memodifikasi pilihan berdasarkan hasil enkripsi sebelumnya. Dalam chosen-plaintext attack, kriptoanalis mungkin hanya dapat memiliki plaintekxt dalam suatu blok besar untuk dienkripsi dalam adaptive-chosen-plaintext attack ini ia dapat memilih blok plaintext yang lebih kecil dan kemudian memilih yang lain berdasarkan hasil yang pertama, proses ini dapat dilakukannya terus menerus hingga ia dapat memperoleh seluruh informasi.

2.7. IP (Internet Protocol)

IP Address adalah angka yang secara unik mengidentifikasi setiap host pada jaringan IP. IP Address beroperasi pada lapisan jaringan tumpukan protokol TCP/IP, jadi mereka berbeda dan terpisah dari alamat MAC lapisan data link tingkat rendah, seperti alamat MAC Ethernet. IP Address adalah 32-bit bilangan biner, yang berarti bahwa secara teoritis, maksimal dari sesuatu di lingkungan 4.000.000.000 alamat host unik dapat ada di seluruh internet. kita berpikir IP address akan cukup digunakan, tetapi TCP/IP di tempat tertentu memiliki pembatasan tentang bagaimana IP Address dialokasikan. Pembatasan ini sangat membatasi jumlah total IP Address yang dapat digunakan. Banyak ahli memprediksi bahwa kita akan segera kehabisan IP Address. Namun, teknik baru untuk bekerja dengan IP Address telah membantu untuk meringankan masalah ini, dan standar untuk 128-bit IP Address telah diadopsi, meskipun masih belum digunakan secara luas.

Ada jenis alamat IP tertentu. Meskipun semua alamat IP terdiri dari angka atau huruf, tidak semua alamat digunakan untuk tujuan yang sama. Ada alamat IP private, alamat IP public, alamat IP statis, dan alamat IP dinamis. Setiap jenis alamat IP dapat menjadi alamat IPv4 atau alamat IPv6. Alamat IP Private Ini digunakan di dalam jaringan, misalnya, jaringan rumah yang digunakan oleh tablet, kamera Wi-Fi, printer nirkabel, dan PC desktop. Jenis alamat IP ini menyediakan cara bagi perangkat untuk berkomunikasi dengan router dan perangkat lain di jaringan rumah pribadi. Alamat IP private dapat diatur secara manual atau ditetapkan secara otomatis oleh router.

Alamat IP Publik digunakan di luar jaringan dan ditetapkan oleh ISP. Ini adalah alamat utama yang digunakan jaringan rumah atau bisnis untuk berkomunikasi dengan seluruh perangkat jaringan di seluruh dunia (misalnya, internet). Ini menyediakan cara bagi perangkat di rumah, misalnya, untuk mencapai ISP, dan oleh karena itu, dunia luar memungkinkan perangkat untuk mengakses situs web dan berkomunikasi langsung dengan komputer dan server lain di seluruh dunia. Alamat IP pribadi dan alamat IP publik bersifat dinamis atau statis, yang berarti bahwa, masing-masing, mereka berubah atau tidak. Alamat IP yang ditetapkan oleh server DHCP adalah alamat IP dinamis. Jika perangkat tidak mengaktifkan DHCP atau tidak mendukung DHCP, maka alamat IP harus ditetapkan secara manual, dalam hal ini perangkat disebut alamat IPstatis. (Ardhiansyah et al., 2020)

2.8. Definisi Python

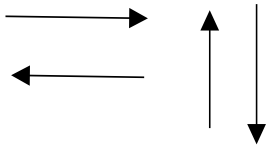
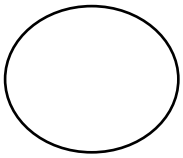
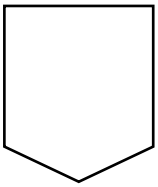
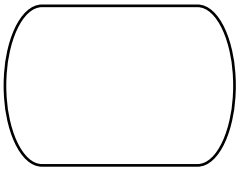
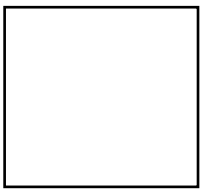
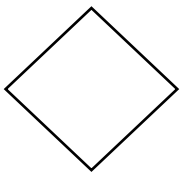
Python adalah salah satu bahasa pemograman tingkat tinggi yang bersifat interpreter, interactive, object-oriented, dan dapat beroperasi hampir di semua platform: Mac, Linux, dan Windows. Python termasuk bahasa pemograman yang mudah dipelajari karena sintaks yang jelas, dapat dikombinasikan dengan penggunaan modul siap pakai, dan struktur data tingkat tinggi yang efisien. Distribusi Python dilengkapi dengan suatu fasilitas seperti shell di Linux. Lokasi penginstalan Python biasa terletak di “/usr/bin/Python”, dan bisa berbeda. Menjalankan Python, cukup dengan mengetikan “Python”, tunggu sebentar lalu muncul tampilan “>>>”, berarti Python telah siap menerima perintah. Ada juga tanda “...” yang berarti baris berikutnya dalam suatu blok prompt '>>>'. Text editordigunakan untuk

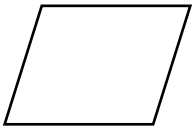
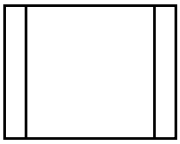
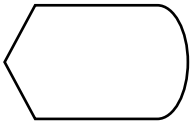
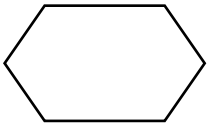
modus skrip. Untuk membangun penelitian ini digunakan wxPython yang merupakan toolkit GUI untuk bahasa pemrograman Python. wxPython memungkinkan programmer Python untuk membuat aplikasi dengan pondasi kuat, grafis antarmuka dengan penggunayang sangat fungsional, sederhana, dan mudah. wxPython diimplementasikan sebagai modul ekstensi oleh Python (kode asli). wxPython membungkus wxWidget sebagai salah satu GUI library populer yang ditulis dalam bahasa C++. Selain itu, digunakan pula Boa Constructor yang merupakan Integrated Development Environment (IDE) untuk Python dan wxPython GUI Builder yang cross-platform. Boa Constructor mampu membuat, memanipulasi frame secara visual (tanpa skrip), dan ada banyak object inspector seperti: browser objek, hirarki warisan, debugger yang canggih, dan bantuan yang sudah terintegrasi. (Ade, 2022)

2.9. Flowchart

Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. Flowchart merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung (Ii & Teori, 2016). Simbol – simbol flowchart:

Tabel 2.9 Simbol *Flowchart*

Simbol	Nama	Fungsi
	*Flow*	Simbol yang digunakan untuk menghubungkan antara simbol yang satu sengan simbol yang lain
	*On-page*	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama
	*Off-page*	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda
	*Terminator*	Simbol yang menyatakan awal atau akhir suatu program
	*Process*	Simbol yang menyatakan suatu proses yang dilakukan computer
	*Decision*	Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu ya dan tidak

	*Input/Output*	Simbol yang menyatakan proses input atau output tanpa tergantung alat
	*Manual Operation*	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer
	*Document*	Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik atau output yang perlu dicetak
	*Predefine Process*	Simbol untuk pelaksanaan suatu bagian (sub-program) atau procedure
	*Display*	Simbol yang menyatakan peralatan output yang digunakan
	*Preperation*	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal

BAB III

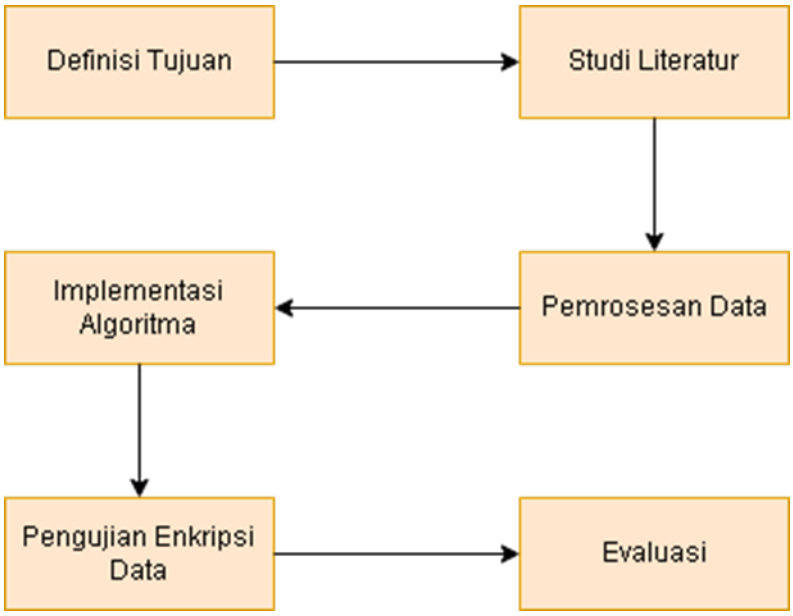
METODOLOGI PENELITIAN

3.1. Metode Penelitian

Metode pengumpulan data yang penulis gunakan didalam penelitian ini adalah observasi dan studi pustaka. Penulis melakukan observasi metode observasi dengan cara membaca beberapa jurnal atau sumber lainnya yang terkait dengan penelitian penulis. Metode observasi adalah suatu teknik pengumpulan data yang dilakukan melalui sesuai dengan pegamatan, dengan di sertai pecatatan-pencatatan tentang keadaan objek terkait dengan sasarannya.

3.2. Prosedur Penelitian

Penelitian ini memiliki beberapa tahap, yaitu defenisi tujuan, studi literatur, pemrosesan data, implementasi algoritma, pengujian dalam mengenkripsi data, serta melakukan pengujian dan evaluasi.



Gambar 3.2 Prosedur Penelitian

3.3. Proses Enkripsi IP dengan Algoritma RSA

1. Persiapan :
    1. Membuat Pasangan Kunci RSA:
        1) Pilih dua bilangan prima besar (p dan q) yang berbeda.
        2) Hitung n = p*q.
        3) Hitung ϕ(n) = (p – 1) * (q – 1).
        4) Pilih bilangan bulat e sedemikian rupa sehingga gcd(e, ϕ(n)) = 1.
        5) Hitung d = e^-1 mod ϕ(n).
        6) Pasangan (e, n) adalah kunci public, dan (d, n) adalah kunci privat.
2. Enkripsi IP:
    1) Ubah alamat IP (dalam bentuk decimal) menjadi bilangan bulat (misalnya dengan konversi biner).
    2) Pastikan bilangan bulat yang dihasilkan lebih kecil dari n.
    3) Lakukan enkripsi dengan rumus:

    Ciphertext = plaintext^e mod n

    Dimana : ciphertext adalah alamat IP terenkripsi, sedangkan plaintext adalah alamat IP dalam bentuk bilangan bulat, e adalah kunci public dan n adalah modulus.
3. Penyimpanan atau Transmisi:
    1) Alamat IP terenkripsi (ciphertext) dapat disimpan atau ditransmisikan ke lokasi lain.
    2) Contoh

    Misalkan alamat IP yang ingin dienkripsi adalah 192.168.1.10.

    Konversi decimal ke biner:

192.168.1.10 = 11000000 10101000 00000001 00001010

Konversi biner ke bilangan bulat:

01100000101010000000000100001010 = 2867440

Anggapkan p = 17 dan q = 23, sehingga n = p*q = 391.

Hitung ϕ(n) = (p – 1) * (q – 1) = 336.

Pilih e = 11, sehingga gcd(e, ϕ(n)) = 1.

Hitung d = e^-1 mod ϕ(n) = 101.

3) Enkripsi:

Ciphertext = 2867440^11 mod 391 = 135203

Alamat IP terenkripsi adalah 135203.

4) Deskripsi:

Untuk mendekripsi alamat IP, penerima harus menggunakan kunci privat (d) dan modulus (n).

Plaintext = ciphertext^d mod n = 135203^101 mod 391 = 1921680010

Konversi bilangan bulat ke biner:

00000000101010000000000100001010 = 11000000 10101000 00000001 00001010

Konversi biner ke decimal:

11000000 10101000 00000001 00001010 = 192.168.1.10

Alamat IP yang didekripsi adalah 192.168.1.10, sama dengan alamat IP asli.

3.4. Flowchart Implementasi Algoritma RSA pada Enkripsi Data

Flowchart implementasi algoritma RSA pada enkripsi data alur ini menggambarkan langkah-langkah yang diperlukan untuk mengetahui cara kerja mengimplementasikan algoritma RSA pada enkripsi data.

Gambar 3.3 Flowchart Implementasi algoritma RSA pada Enkripsi data

Adapun sistematika *flowchart* diatas yaitu sebagai berikut:

1. Mulai: Proses dimulai.
2. Mengumpulkan data lokasi berdasarkan alamat IP: Data lokasi dikumpulkan berdasarkan alamat IP.
3. Menggunakan algoritma RSA untuk enkripsi data lokasi: Data lokasi yang telah dikumpulkan kemudian dienkripsi menggunakan algoritma RSA.
4. Memanipulasi lokasi berdasarkan data yang telah dienkripsi: Data lokasi yang telah dienkripsi diproses dan dimanipulasi.

5. Menyimpan data yang telah dimanipulasi: Data yang telah dimanipulasi dan tetap dalam keadaan terenkripsi disimpan dalam system penyimpanan.
6. Selesai: Titik akhir dari proses ini.

3.5. Implementasi Algoritma RSA

Dari sekian banyak algoritma kriptografi kunci publik yang pernah dibuat, algoritma yang paling populer adalah algoritma RSA. Algoritma ini melakukan pemfaktoran bilangan yang sangat besar. Oleh karena alasan tersebut RSA dianggap aman. Untuk membangkitkan dua kunci, dipilih dua bilangan prima acak yang besar. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (Massachussets Institute of Technology) pada tahun 1976, yaitu : Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. Kekuatan algoritma RSA terletak pada sulitnya memfaktorkan bilangan besar menjadi faktor-faktor bilangan primanya, sehingga semakin besar bilangan prima yang digunakan semakin baik atau aman. Dalam kriptografi menggunakan algoritma RSA terdapat tiga proses yaitu proses pembangkitan kunci publik dan kunci privat, proses enkripsi, dan proses dekripsi.

Enkripsi : $C = P^e \ mod \ n$

Deskripsi : $P = C^d \ mod \ n$

3.6. Jadwal Penelitian

Setiap rancangan pada penelitian pastinya perlu dilengkapi dengan jadwal yang sudah dilakukan. Berikut ini adalah rincian penilaian.

Tabel 3.5 Jadwal Penelitian

Kegiatan Penelitian	Bulan					
	Feb	Mar	Apr	Mei	Jun	Jul
1. Persiapan Penelitian						
a. Pengajuan Judul						
b. Pengajuan SK Pembimbing						
c. Observasi						
d. Penyesunan Proposal						
e. Seminar Proposal						
2. Implementasi & Pengumpulan Data						
a. Pembuatan Sistem						
b. Pengumpulan Data						
3. Pemrosesan Data Dan Pelaporan						
a. Validasi dan Hasil						
b. Penyusunan Laporan Skirpsi						

BAB IV

HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data Geolokasi

Pengumpulan data geolokasi dilakukan dengan mengumpulkan alamat IP dari beberapa sumber yang dapat dipercaya. Sumber-sumber ini termasuk layanan publik yang menyediakan informasi alamat IP serta basis data internal yang ada. Teknik geolokasi berbasis alamat IP kemudian diterapkan untuk mengubah alamat IP tersebut menjadi data geolokasi yang mencakup informasi seperti koordinat geografis (lintang dan bujur), kota, negara, dan penyedia layanan internet (ISP).

Proses geolokasi dilakukan dengan menggunakan API atau perangkat lunak khusus yang memetakan alamat IP ke lokasi geografis. Hasil dari proses ini adalah sekumpulan data geolokasi yang mencerminkan posisi geografis dari alamat IP yang dikumpulkan. Data ini kemudian disimpan dalam format terstruktur untuk memfasilitasi proses enkripsi lebih lanjut.

Analisis awal dari data yang dikumpulkan menunjukkan variasi dalam keakuratan dan detail informasi geolokasi yang diperoleh. Beberapa alamat IP dapat menghasilkan data lokasi yang sangat tepat, sementara yang lain mungkin hanya menunjukkan lokasi umum seperti kota atau negara. Namun demikian, data yang diperoleh cukup representatif untuk tujuan penelitian ini, yaitu untuk menguji implementasi algoritma RSA dalam mengenkripsi data geolokasi.

Selama proses pengumpulan data, perhatian khusus diberikan pada validitas dan legalitas sumber data untuk memastikan bahwa informasi yang dikumpulkan adalah sah dan dapat digunakan tanpa melanggar aturan atau ketentuan yang berlaku. Data yang terkumpul kemudian diproses lebih lanjut untuk persiapan enkripsi, termasuk pembersihan data untuk menghapus entri yang tidak lengkap

atau tidak akurat. Dengan terkumpulnya data geolokasi ini, maka penelitian dapat dilanjutkan ke tahap berikutnya, yaitu implementasi algoritma RSA untuk mengenkripsi data tersebut. Data geolokasi yang telah disiapkan akan digunakan sebagai input untuk proses enkripsi, memastikan bahwa seluruh proses didasarkan pada data yang valid dan relevan.

4.2. Perhitungan Manual dan Menggunakan Python Pada Algoritma RSA Berdasarkan IP

Pada perhitungan manual algoritma RSA berdasarkan IP ini akan menjelaskan bagaimana cara melakukan perhitungan manual algoritma RSA berdasarkan IP secara spesifik. Mulai dari pembangkitan kunci, proses enkripsi dan proses deskripsi.

1. Proses Enkripsi dan Deskripsi IP Dengan Algoritma RSA

a. Membuat Pasangan Kunci RSA:

   1) Pilih dua bilangan prima besar (p dan q) yang berbeda.
   2) Hitung n = p*q.
   3) Hitung φ(n) = (p – 1) * (q – 1).
   4) Pilih bilangan bulat e sedemikian rupa sehingga gcd(e, φ(n)) = 1.
   5) Hitung d = e^-1 mod φ(n).
   6) Pasangan (e, n) adalah kunci public, dan (d, n) adalah kunci privat.

2. Enkripsi IP:

   1) Ubah alamat IP (dalam bentuk decimal) menjadi bilangan bulat (misalnya dengan konversi biner).
   2) Pastikan bilangan bulat yang dihasilkan lebih kecil dari n.
   3) Lakukan enkripsi dengan rumus:

Ciphertext = plaintext^e mod n

Dimana : ciphertext adalah alamat IP terenkripsi, sedangkan plaintext adalah alamat IP dalam bentuk bilangan bulat, e adalah kunci public dan n adalah modulus.

3. Proses Perhitungan Manual Alamat IP Dengan Algoritma RSA:

   Alamat IP yang ingin dienkripsi adalah 114.122.5.15

   Pertama, kita akan menkonversi setiap bagian dari IP ke biner:

114	01110010
122	01111010
5	00000101
15	00001111

Lalu kita akan menggabungkan semuanya menjadi satu:

01110010 01111010 00000101 00001111

Lalu kita akan menkonversikan biner ini ke decimal:

01110010 01111010 00000101 00001111 = 192.059.7311

Lalu kita akan menkonversi alamat IP ke bilangan bulat

Alatmat IP 114.122.5.15 kita akan mengonversinya menjadi satu bilangan bulat.

Cara konversi:

1) Pisahkan setiap octet dari alamat IP.
2) Kalikan setiap octet dengan pangkat 256 yang sesuai, dan kemudian jumlah hasilnya.

Contoh:

a. Alamat IP. 114.122.5.15

b. Oktet 114, 122, 5, 15

Perhitungannya:

IP_to_int = $(114 \times 256^3) + (122 \times 256^2) + (5 \times 256) + 15$

Langkah-langkah:

1. $114 \times 256^3 = 114 \times 16777216 = 1912602624$
2. $122 \times 256^2 = 122 \times 65536 = 7995392$
3. $5 \times 256 = 1280$
4. $15 = 15$

Jumlahkan semuanya:

IP_to_int = 1912602624 + 7995392 + 1280 + 15 = 1920597311

4. Langkah-langkah RSA

1) Langkah 1 yaitu memilih nilai p dan q

a. p = 17

b. q = 23

2) Langkah 2 yaitu menghitung n

a. $n = p \times q$

b. $n = 17 \times 23 = 391$

3) Langkah 3 yaitu menghitung $\phi(n)$

a. $\phi(n) = (p - 1) \times (q - 1)$

b. $\phi(n) = (17 - 1) \times (23 - 1) = 16 \times 22 = 352$

4) Langkah 4 memilih e

a. Pilih e yang gcd (e, $\phi(n)$) = 1

b. e = 13 (Sudah dipastikan coprime dengan 352)

5) Langkah 5 menghitung d

a. Menggunakan Extended Euclidean Algorithm untuk menemukan d:

b. $d = e^{-1} \bmod \phi(n)$

c. $13d = 1 \bmod 352$

d. Menggunakan algoritma Euclidean Extended:

a) $352 = 27 \times 13 + 1$

b) $1 = 352 - 27 \times 13$

c) $d = 325$

d) Jadi $d = 325$

5. Proses Enkripsi

1. Alamat IP 114.122.5.15 kita konversi menjadi decimal (disini saya menggunakan 114 untuk dalam perhitungan enkripsi ini):
2. Plaintext = 114
3. Rumus enkripsi: Ciphertext = plaintext^e mod n
4. Ciphertext = $114^{13} \bmod 391$

Perhitungan menggunakan metode eksponensiasi modular untuk menghitung $114^{13} \bmod 391$:

1. $114^1 \bmod 391 = 114$
2. $114^2 \bmod 391 = 12996 \bmod 391 = 93$
3. $114^4 \bmod 391 = 93^2 \bmod 391 = 8649 \bmod 391 = 47$
4. $114^8 \bmod 391 = 47^2 \bmod 391 = 2209 \bmod 391 = 254$
5. $114^{13} \bmod 391 = 114 \times 93 \times 47 \times 254 \bmod 391$

Gabungkan hasil dengan eksponen dalam bentuk biner (1101):

$114^{13} = 114^{8+4+1} = 114^8 . 114^4 . 114 \bmod 391$

$114^{13} = 254.47.114 \bmod 391$

1. 254 × 47 mod 391 = 11938 mod 391 = 11938 – 30 × 391 = 208
2. 208 × 114 mod 391 = 23712 mod 391 = 23712 – 60 × 391 = 252

Jadi, ciphertext = 252

6. Proses Deskripsi
    1. Rumus deskripsi: plaintext = ciphertext^d mod n
    2. Plaintext 252^{325} mod 391

Menggunakan Metode eksponensiasi modular untuk melalukan perhitungan.

1. 252^{1} mod 391 = 252
2. 252^{2} mod 391 = 63504 mod 391 = 162
3. 252^{4} mod 391 = 162^{2} mod 391 =26244 mod 391 = 47
4. 252^{8} mod 391 = 47^{2} mod 391 = 2209 mod 391 = 254
5. 252^{16} mod 391 = 254^{2} mod 391 = 64516 mod 391 = 101
6. 252^{32} mod 391 = 101^{2} mod 391 = 10201 mod 391 = 35
7. 252^{64} mod 391 = 35^{2} mod 391 = 1225 mod 391 = 52
8. 252^{128} mod 391 = 52^{2} mod 391 = 2704 mod 391 = 358
9. 252^{256} mod 391 = 358^{2} mod 391 = 128164 mod 391 = 270

Menggabungkan hasil

$252^{325} = 252^{256}. 252^{64}. 252^{4}. 252^{1}$ mod 391

1. 358 × 35 mod 391 = 12530 mod 391 = 12530 – 32 × 391 = 18
2. 18 × 8 mod 391 = 144 mod 391 = 144 – 0 × 391 = 144

Jadi, plaintext = 114

Kesimpulan

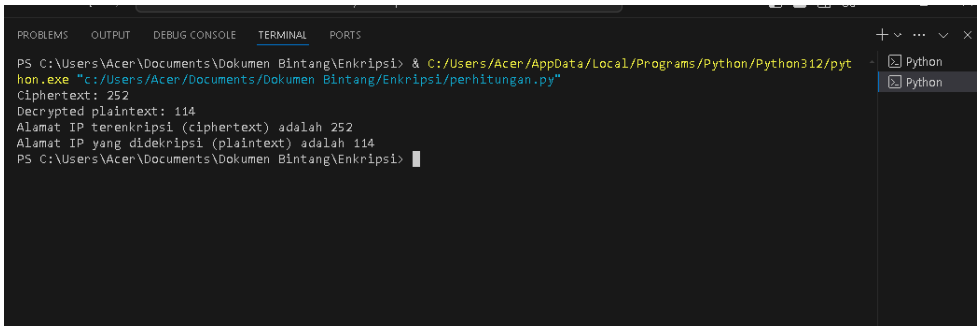
Setelah melakukan perhitungan manual dan telah mendapatkan hasil perhitungan dengan metode eksponensiasi modular yang tepat maka hasilnya:

1. Ciphertext 252
2. Plaintext 114

Setelah melakukan perhitungan manual disini kita mencoba melakukan perhitungan dengan menggunakan bahasa pemograman python untuk mendapatkan hasil yang serupa seperti pada perhitungan manual diatas yaitu sebagai berikut:

```python
def modular_exponentiation(base, exp, mod):
    result = 1
    base = base % mod
    while exp > 0:
        if exp % 2 == 1:
            result = (result * base) % mod
        exp = exp >> 1
        base = (base * base) % mod
    return result

p = 17
q = 23
n = p * q
phi_n = (p - 1) * (q - 1)
e = 13

def mod_inverse(e, phi):
    t, newt = 0, 1
    r, newr = phi, e
    while newr != 0:
        quotient = r // newr
        t, newt = newt, t - quotient * newt
        r, newr = newr, r - quotient * newr
    if r > 1:
        raise ValueError("e tidak memiliki inverse modulo ϕ(n)")
    if t < 0:
        t = t + phi
    return t
```

```
d = mod_inverse(e, phi_n)
plaintext = 114

ciphertext = modular_exponentiation(plaintext, e, n)
print(f"Ciphertext: {ciphertext}")

decrypted_plaintext = modular_exponentiation(ciphertext, d, n)
print(f"Decrypted plaintext: {decrypted_plaintext}")

print(f"Alamat IP terenkripsi (ciphertext) adalah {ciphertext}")
print(f"Alamat IP yang didekripsi (plaintext) adalah
{decrypted_plaintext
```



Gambar 4.2 Hasil Perhitungan Ekripsi dan Deskripsi

Pada gambar 4.3 menampilkan hasil gambar yaitu sebagai berikut:

1. Ciphertext: 252
2. Decrypted plaintext: 114
3. Alamat IP yang terenkripsi (ciphertext) adalah 252
4. Alamat IP yang dideskripsi (plaintext) adalah 114

Kesimpulan

Setelah melakukan perhitungan dengan menggunakan python dan telah mendapatkan hasil perhitungan yang serupa dengan hasil perhitungan manual maka hasilnya pada pemograman python ini yaitu:

1. Ciphertext 252
2. Plaintext 114

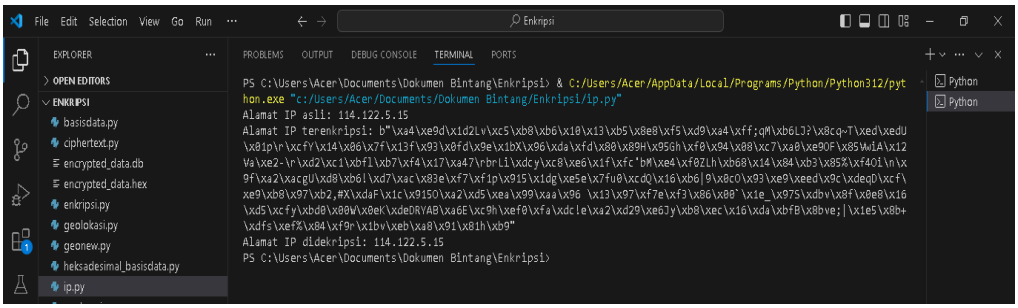
4.3. Implementasi Enkripsi Pada Alamat IP Dengan Algoritma RSA

Implementasi ini dilakukan untuk dapat menenkripsikan alamat IP dengan menggunakan algoritma RSA. RSA merupakan algoritma enkripsi asimetris yang menggunakan sepasang kunci, yaitu kunci public dan kunci privat, untuk proses enkripsi dan deskripsi data. Implementasi ini dilakukan bertujuan untuk dapat melindungi informasi alamat IP dari akses yang tidak sah. Adapun hasil implementasi enkripsi dan deskripsinya menggunakan bahasa pemograman Python menggunakan Visual Studio Code yaitu sebagai berikut:

```
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives import serialization

def generate_rsa_keypair():
    """
    Membuat pasangan kunci publik dan privat RSA.

    Returns:
        Tuple: (kunci_publik, kunci_privat)
    """
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=2048
    )
    public_key = private_key.public_key()
    return public_key, private_key

def encrypt_ip(ip_address, public_key):
    """
    Enkripsi alamat IP dengan kunci publik RSA.

    Args:
        ip_address (str): Alamat IP yang ingin dienkripsi.
        public_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPublicKey): Kunci
publik RSA.

    Returns:
```

```
        bytes: Alamat IP terenkripsi.
    """
    ip_bytes = ip_address.encode('ascii')
    ciphertext = public_key.encrypt(
        ip_bytes,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return ciphertext

def decrypt_ip(ciphertext, private_key):
    """
    Dekripsi alamat IP terenkripsi dengan kunci privat RSA.

    Args:
        ciphertext (bytes): Alamat IP terenkripsi.
        private_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPrivateKey): Kunci
privat RSA.

    Returns:
        str: Alamat IP yang didekripsi.
    """
    plaintext = private_key.decrypt(
        ciphertext,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return plaintext.decode('ascii')

# Contoh penggunaan
public_key, private_key = generate_rsa_keypair()

ip_address = "114.122.5.15"
encrypted_ip = encrypt_ip(ip_address, public_key)
decrypted_ip = decrypt_ip(encrypted_ip, private_key)

print("Alamat IP asli:", ip_address)
print("Alamat IP terenkripsi:", encrypted_ip)
print("Alamat IP didekripsi:", decrypted_ip)
```

Maka hasil pengujian dari sourcode di atas menampilkan hasil sebagai berikut:



Gambar 4.3 Proses Enkripsi Pada IP

Pada gambar 4.3 di atas menampil hasil yaitu sebagai berikut:

1. Alamat IP asli: 114.122.5.15
2. Alamat IP terenkripsi:
3. b"\xa4\xe9d\x1d2Lv\xc5\xb8\xb6\x10\x13\xb5\x8e8\xf5\xd9\xa4\xff;qM\xb6LJ?\x8cq~T\xed\xedU\x01p\r\xcfY\x14\x06\x7f\x13f\x93\x0fd\x9e\x1bX\x96\xda\xfd\x80\x89H\x95Gh\xf0\x94\x08\xc7\xa0\xe9OF\x85VwiA\x12Va\xe2\r\xd2\xc1\xbfl\xb7\xf4\x17\xa47\rbrLi\xdcy\xc8\xe6\x1f\xfc'bM\xe4\xf0ZLh\xb68\x14\x84\xb3\x85%\xf4Oi\n\x9f\xa2\xacgU\xd8\xb6!\xd7\xac\x83e\xf7\xf1p\x915\x1dg\xe5e\x7fu0\xcdQ\x16\xb6|9\x0cO\x93\xe9\xeed\x9c\xdeqD\xcf\xe9\xb8\x97\xb2,#X\xdaF\x1c\x915O\xa2\xd5\xea\x99\xaa\x96\x13\x97\xf7e\xf3\x86\x00`\x1e_\x97S\xdbv\x8f\x0e8\x16\xd5\xcfy\xbd0\x00W\x0eK\xdeDRYAB\xa6E\xc9h\xef0\xfa\xdc!e\xa2\xd29\xe6Jy\xb8\xec\x16\xda\xbfB\x8bve;|\x1e5\x8b+\xdfs\xef%\x84\xf9r\x1bv\xeb\xa8\x91\x81h\xb9"
4. Alamat IP dideskripsi: 114.122.5.15

Implementasi enkripsi pada alamat IP dengan algoritma RSA sudah berhasil dilakukan. Algoritma RSA dapat memberikan keamanan tambahan pada alamat IP dengan mengubahnya menjadi format terenkripsi yang hanya dapat dideskripsi

dengan kunci privat yang sesuai. Penggunaan kunci publik dan kunci privat memastikan bahwa hanya pihak yang memiliki kunci privat yang dapat mengakses informasi yang asli dari alamat IP yang sudah terenkripsi.

4.4. Enkripsi Data Geolokasi Berdasarkan Alamat IP Menggunakan Python

Data geolokasi dilakukan dan dipersiapkan, langkah berikutnya adalah mengenkripsi data tersebut dan memastikan bahwa data geolokasi dalam format tepat dan siap untuk dienkripsi. Proses enkripsi ini dimulai dengan memastikan bahwa data geolokasi dalam format yang tepat dan siap untuk dienkripsi. Data yang siap dienkripsi mencakup informasi seperti koordinat geografis (lintang dan bujur), kota, negara, dan penyedia layanan internet (ISP).

Data geolokasi dienkripsi menggunakan kunci publik RSA yang telah dibuat sebelumnya. Ini menghasilkan bentuk data terenkripsi yang tidak dapat dibaca atau dipahami tanpa kunci dekripsi yang tepat. Implementasi enkripsi dapat dilakukan dengan menggunakan pustaka kriptografi yang tersedia, seperti pustaka kriptografi dalam bahasa pemrograman Python. Pustaka ini menerima data geolokasi dan kunci publik sebagai input dan menghasilkan ciphertext sebagai output. Adapun hasil implementasi enkripsi dan deskripsinya menggunakan bahasa pemograman Python menggunakan Visual Studio Code yaitu sebagai berikut:

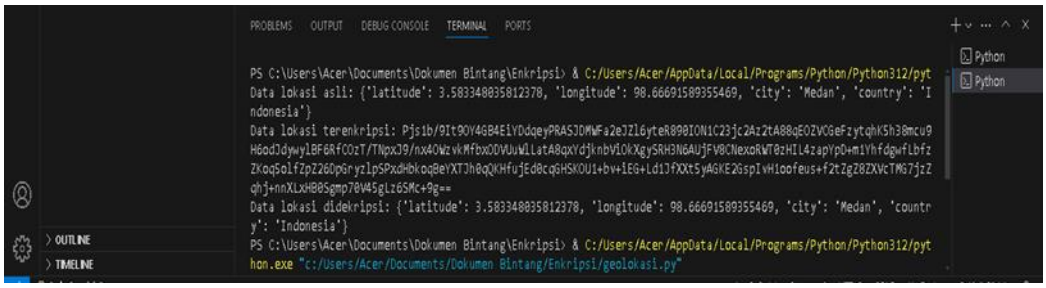
```
import json
import requests
import base64
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import serialization, hashes

def generate_rsa_keypair():
    """
    Membuat pasangan kunci publik dan privat RSA.
```

```
    Returns:
        Tuple: (kunci_publik, kunci_privat)
    """
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=2048
    )
    public_key = private_key.public_key()
    return public_key, private_key

def get_location_data(ip_address):
    """
    Mengambil data lokasi berdasarkan alamat IP.

    Args:
        ip_address (str): Alamat IP.

    Returns:
        dict: Data lokasi (latitude, longitude, city, country).
    """
    url = f"https://freegeoip.app/json/{ip_address}"
    response = requests.get(url)
    if response.status_code == 200:
        data = json.loads(response.text)
        if "latitude" in data and "longitude" in data:
            return {
                "latitude": data["latitude"],
                "longitude": data["longitude"],
                "city": data.get("city"),
                "country": data.get("country_name")
            }
        else:
            return {}
    else:
        return {}

def encrypt_data(data, public_key):
    """
    Enkripsi data JSON dengan kunci publik RSA.

    Args:
        data (dict): Data JSON yang ingin dienkripsi.
        public_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPublicKey): Kunci publik
RSA.

    Returns:
```

```
        str: Data terenkripsi dalam format Base64.
    """
    # Konversi data JSON ke bytes
    data_bytes = json.dumps(data).encode('utf-8')

    # Enkripsi data dengan kunci publik RSA
    ciphertext = public_key.encrypt(
        data_bytes,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    # Encode hasil enkripsi ke Base64
    encoded_ciphertext = base64.b64encode(ciphertext)
    return encoded_ciphertext.decode('ascii')

def decrypt_data(encoded_ciphertext, private_key):
    """
    Dekripsi data terenkripsi dengan kunci privat RSA.

    Args:
        encoded_ciphertext (str): Data terenkripsi dalam format Base64.
        private_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPrivateKey): Kunci privat
RSA.

    Returns:
        dict: Data yang didekripsi.
    """
    # Decode hasil enkripsi dari Base64
    ciphertext = base64.b64decode(encoded_ciphertext)

    # Dekripsi data dengan kunci privat RSA
    plaintext = private_key.decrypt(
        ciphertext,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    try:
        return json.loads(plaintext.decode('utf-8'))
    except Exception:
        return {}
```

```
# Contoh penggunaan
public_key, private_key = generate_rsa_keypair()

ip_address = "114.122.5.15"

location_data = get_location_data(ip_address)
if location_data:
    encrypted_data = encrypt_data(location_data, public_key)
    decrypted_data = decrypt_data(encrypted_data, private_key)

    print("Data lokasi asli:", location_data)
    print("Data lokasi terenkripsi:", encrypted_data)
    print("Data lokasi didekripsi:", decrypted_data)
else:
    print("Gagal mendapatkan data lokasi.")
```

Maka hasil dari pengujian dari sourcode diatas menampilkan hasil sebagai berikut:



Gambar 4.4 Proses Enkripsi dan Deskripsi Geolokasi

Pada gambar 4.4 di atas menampilkan hasil yaitu sebagai berikut:

1. Data lokasi asli: {'latitude':3.583348035812378, 'longitude': 98.66691589355469, 'city': 'Medan', 'country': 'Indonesia'}
2. Data lokasi terenkripsi:

   Rh+EdzCdgx6qzfCVDgQUuOzxrsDxy2urvp6IgHiZbmg4dR3uh0jDyBxB
   5ZmCSKOCqFNet4PJmmPtihVKcTiU2pVE78ig2p3GfwjzB1WoVIbe8M
   7JopYLkMLnG2Bja0RtR08BTjQmA1A13LlaRU5QrDD3IQxb/gQWkDp
   SBgxyTmBfCikK1+uNplMLqbPAHHFbIiP434y9cTvJkTfuIjUZL+PgDO

TjoDCdiwqa4IyVXyqrpjiIp/62FDJ9HqXOmfxkV1Hw4+gKSs380ho6uDo
/Gsgm8B+ewx2cOGDNKJumhB4pLGInWJnuTNY1+20E9NMCy1dXm8
RhvnxCnY2sHLGFxw==

3. Data lokasi dideskripsi: {'latitude':3.583348035812378, 'longitude': 98.66691589355469, 'city': 'Medan', 'country': 'Indonesia'}

Hasil dari proses enkripsi adalah sekumpulan ciphertext yang mewakili data geolokasi dalam bentuk terenkripsi. Analisis awal terhadap ciphertext menunjukkan bahwa data asli tidak dapat dikenali atau dipulihkan tanpa kunci privat yang sesuai, menandakan bahwa proses enkripsi berjalan dengan efektif.

4.5. Penyimpanan Data Terenkripsi Kedalam Basis Data

Setelah melakukan proses enkripsi dan deskripsi selesai dan data geolokasi telah diubah menjadi ciphertext, langkah selanjutnya adalah menyimpan data terenkripsi ini dengan aman ke dalam basis data. Penyimpanan data terenkripsi ini dilakukan karena dengan tujuan untuk dapat menjaga integritas dan kerahasiaan data, serta dapat memastikan bahwa hanya pihak yang memiliki akses yang sah dapat mengakses data tersebut.

Data yang sudah terenkripsi disimpan dalam basis data yang dirancang untuk keamanan data. Basis data ini dapat berupa sistem manajemen basis data relasional (RDBMS) seperti MySQL atau PostgreSQL. Disini saya menggunakan basis data SQLite untuk dapat menyimpan data yang sudah terenkripsi, SQLite merupakan basis data yang tidak memerlukan server terpisah untuk menjalankan fungsinya, berbeda dengan MySQL atau PostgreSQL yang dimana memerlukan server. Adapun hasil pembuatan basis data ini menggunakan sourcode Python

dengan Visual Studio code, maka sourcode akan ditampilkan yaitu sebagai berikut:

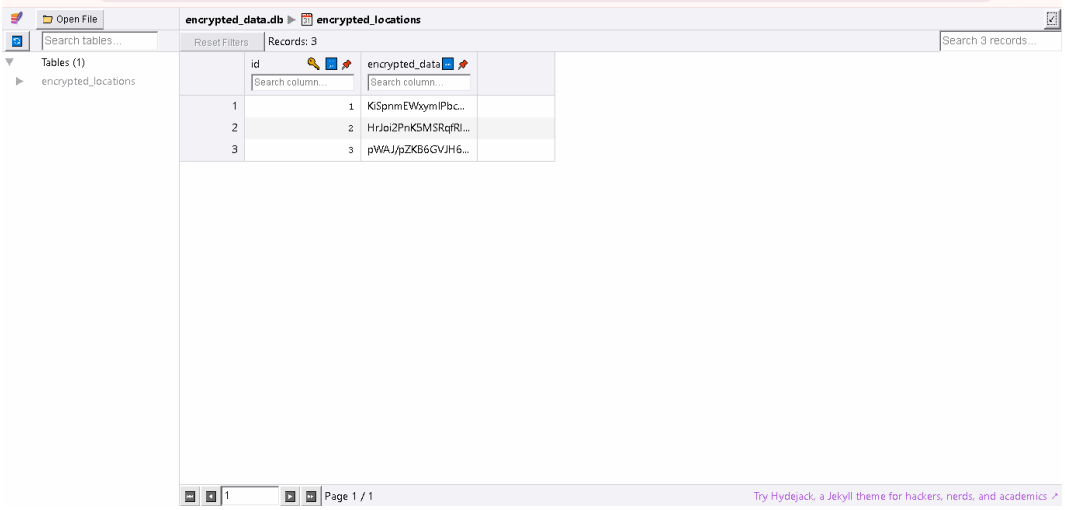
```
import json
import requests
import base64
import sqlite3
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import serialization, hashes

def generate_rsa_keypair():
    """
    Membuat pasangan kunci publik dan privat RSA.

    Returns:
        Tuple: (kunci_publik, kunci_privat)
    """
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=2048
    )
    public_key = private_key.public_key()
    return public_key, private_key

def get_location_data(ip_address):
    """
    Mengambil data lokasi berdasarkan alamat IP.

    Args:
        ip_address (str): Alamat IP.

    Returns:
        dict: Data lokasi (latitude, longitude, city, country).
    """
    url = f"https://freegeoip.app/json/{ip_address}"
    response = requests.get(url)
    if response.status_code == 200:
        data = json.loads(response.text)
        if "latitude" in data and "longitude" in data:
            return {
                "latitude": data["latitude"],
                "longitude": data["longitude"],
                "city": data.get("city"),
                "country": data.get("country_name")
            }
        else:
```

```
            return {}
    else:
        return {}

def encrypt_data(data, public_key):
    """
    Enkripsi data JSON dengan kunci publik RSA.

    Args:
        data (dict): Data JSON yang ingin dienkripsi.
        public_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPublicKey): Kunci
publik RSA.

    Returns:
        str: Data terenkripsi dalam format Base64.
    """
    # Konversi data JSON ke bytes
    data_bytes = json.dumps(data).encode('utf-8')

    # Enkripsi data dengan kunci publik RSA
    ciphertext = public_key.encrypt(
        data_bytes,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    # Encode hasil enkripsi ke Base64
    encoded_ciphertext = base64.b64encode(ciphertext)
    return encoded_ciphertext.decode('ascii')

def decrypt_data(encoded_ciphertext, private_key):
    """
    Dekripsi data terenkripsi dengan kunci privat RSA.

    Args:
        encoded_ciphertext (str): Data terenkripsi dalam format
Base64.
        private_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPrivateKey): Kunci
privat RSA.

    Returns:
        dict: Data yang didekripsi.
    """
```

```
    # Decode hasil enkripsi dari Base64
    ciphertext = base64.b64decode(encoded_ciphertext)

    # Dekripsi data dengan kunci privat RSA
    plaintext = private_key.decrypt(
        ciphertext,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    try:
        return json.loads(plaintext.decode('utf-8'))
    except Exception:
        return {}

def save_to_database(encrypted_data):
    """
    Menyimpan data terenkripsi ke dalam basis data SQLite.

    Args:
        encrypted_data (str): Data terenkripsi dalam format Base64.
    """
    conn = sqlite3.connect('encrypted_data.db')
    cursor = conn.cursor()

    # Membuat tabel jika belum ada
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS encrypted_locations (
            id INTEGER PRIMARY KEY,
            encrypted_data TEXT
        )
    ''')

    # Menyimpan data terenkripsi ke tabel
    cursor.execute('''
        INSERT INTO encrypted_locations (encrypted_data)
        VALUES (?)
    ''', (encrypted_data,))
    conn.commit()
    conn.close()

def load_from_database(record_id):
    """
    Mengambil data terenkripsi dari basis data SQLite berdasarkan
ID.
```

```
    Args:
        record_id (int): ID dari data terenkripsi.

    Returns:
        str: Data terenkripsi dalam format Base64.
    """
    conn = sqlite3.connect('encrypted_data.db')
    cursor = conn.cursor()

    cursor.execute('SELECT encrypted_data FROM encrypted_locations
WHERE id = ?', (record_id,))
    encrypted_data = cursor.fetchone()
    conn.close()

    if encrypted_data:
        return encrypted_data[0]
    else:
        return None

# Contoh penggunaan
public_key, private_key = generate_rsa_keypair()

ip_address = "114.122.5.15"
location_data = get_location_data(ip_address)
if location_data:
    encrypted_data = encrypt_data(location_data, public_key)
    save_to_database(encrypted_data)

    # Misalkan kita mengambil data dengan ID 1
    encrypted_data_from_db = load_from_database(1)
    if encrypted_data_from_db:
        decrypted_data = decrypt_data(encrypted_data_from_db,
private_key)

        print("Data lokasi asli:", location_data)
        print("Data lokasi terenkripsi:", encrypted_data)
        print("Data lokasi didekripsi:", decrypted_data)
    else:
        print("Data tidak ditemukan di basis data.")
else:
    print("Gagal mendapatkan data lokasi.")
```

Maka hasil dari sourcode di atas dapat menampil enkrip data ke dalam basis data SQLite yaitu sebagai berikut:



Gambar 4.5 Basis Data SQLite Enkripsi Data

Penyimpanan data base ini dilakukan untuk memastikan bahwa data terenkripsi dapat disimpan dan diambil tanpa kehilangan integritas atau keamanannya. Pengunjian menunjukkan bahwa data yang terenkripsi dan disimpan dapat dideskripsi kembali dengan benar, hasil dari pengujian penyimpanan data terenkripsi ini juga dapat menunjukkan bahwa sistem penyimpanan berfungsi dengan baik dan aman. Ciphertext tetap utuh selama penyimpanan, dan hanya pihak yang memiliki file enkripsi yang dapat mengakses atau mengubah data tersebut.

Dengan demikian hasil pengujian basis data ini semoga dapat menjadi peran penting dalam keseluruhan sistem keamanan data, dan dapat memberikan jaminan bahwa data geolokasi yang terenkripsi tidak dapat disalahgunakan atau diakses oleh pihak yang tidak berwenang.

4.6. Pengujian Keamanan

Pengujian keamanan ini dilakukan untuk dapat memastikan bahwa algoritma RSA yang digunakan dalam enkripsi data geolokasi dapat mampu melindungi data dari berbagai jenis serangan. Fokus pengujian ini adalah

mencoba mensimulasikan pada ketahanan terhadap serangan brute force, yang merupakan salah satu metode umum yang sering digunakan untuk mencoba memecahkan enkripsi. Untuk dapat mensimulasikannya, disini saya menggunakan bahasa pemograman python dengan Visual Studio Code.

- Pengujian Brute Force

  Serangan brute force adalah metode di mana penyerang mencoba setiap kemungkinan kombinasi kunci hingga menemukan kunci yang tepat untuk mendeskripsikan data. Dengan menggunakan kunci RSA sepanjang 2048-bit, jumlah kemungkinan kombinasi kunci sangat besar, sehingga serangan brute force menjadi tidak praktis dan dapat memakan waktu. Ini adalah sebuah simulasi pada serangan brute force pada bahasa pemograman python.

```
import json
import requests
import base64
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import serialization, hashes
import time

def generate_rsa_keypair():
    """
    Membuat pasangan kunci publik dan privat RSA.

    Returns:
        Tuple: (kunci_publik, kunci_privat)
    """
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=2048
    )
    public_key = private_key.public_key()
    return public_key, private_key
```

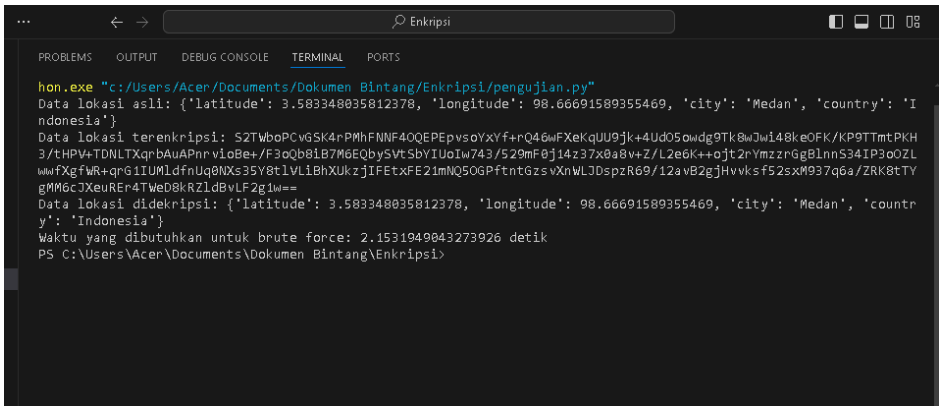
```
def get_location_data(ip_address):
    """
    Mengambil data lokasi berdasarkan alamat IP.

    Args:
        ip_address (str): Alamat IP.

    Returns:
        dict: Data lokasi (latitude, longitude, city, country).
    """
    url = f"https://freegeoip.app/json/{ip_address}"
    response = requests.get(url)
    if response.status_code == 200:
        data = json.loads(response.text)
        if "latitude" in data and "longitude" in data:
            return {
                "latitude": data["latitude"],
                "longitude": data["longitude"],
                "city": data.get("city"),
                "country": data.get("country_name")
            }
        else:
            return {}
    else:
        return {}

def encrypt_data(data, public_key):
    """
    Enkripsi data JSON dengan kunci publik RSA.

    Args:
        data (dict): Data JSON yang ingin dienkripsi.
        public_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPublicKey): Kunci
publik RSA.

    Returns:
        str: Data terenkripsi dalam format Base64.
    """
    # Konversi data JSON ke bytes
    data_bytes = json.dumps(data).encode('utf-8')

    # Enkripsi data dengan kunci publik RSA
    ciphertext = public_key.encrypt(
        data_bytes,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
```

```
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    # Encode hasil enkripsi ke Base64
    encoded_ciphertext = base64.b64encode(ciphertext)
    return encoded_ciphertext.decode('ascii')

def decrypt_data(encoded_ciphertext, private_key):
    """
    Dekripsi data terenkripsi dengan kunci privat RSA.

    Args:
        encoded_ciphertext (str): Data terenkripsi dalam format
Base64.
        private_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPrivateKey): Kunci
privat RSA.

    Returns:
        dict: Data yang didekripsi.
    """
    # Decode hasil enkripsi dari Base64
    ciphertext = base64.b64decode(encoded_ciphertext)

    # Dekripsi data dengan kunci privat RSA
    plaintext = private_key.decrypt(
        ciphertext,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    try:
        return json.loads(plaintext.decode('utf-8'))
    except Exception:
        return {}

def brute_force_rsa(encrypted_data, public_key):
    """
    Simulasi serangan brute force terhadap enkripsi RSA.

    Args:
        encrypted_data (str): Data terenkripsi dalam format Base64.
```

```
        public_key
(cryptography.hazmat.primitives.asymmetric.rsa.RSAPublicKey): Kunci
publik RSA.

    Returns:
        None
    """
    encrypted_bytes = base64.b64decode(encrypted_data)
    start_time = time.time()

    # Mulai simulasi brute force
    for i in range(1, 100000):  # Rentang ini hanya untuk contoh;
brute force RSA nyata membutuhkan rentang yang jauh lebih besar
        try:
            # Coba dekripsi dengan kunci privat yang dihasilkan dari
i
            private_key_candidate = rsa.RSAPrivateNumbers(
                p=i, q=i, d=i, dmp1=i, dmq1=i, iqmp=i,
                public_numbers=rsa.RSAPublicNumbers(
                    e=65537, n=public_key.public_numbers().n
                )
            ).private_key()

            # Coba dekripsi pesan terenkripsi
            plaintext = private_key_candidate.decrypt(
                encrypted_bytes,
                padding.OAEP(
                    mgf=padding.MGF1(algorithm=hashes.SHA256()),
                    algorithm=hashes.SHA256(),
                    label=None
                )
            )

            print("Kunci ditemukan:", i)
            print("Pesan:", plaintext.decode('utf-8'))
            break
        except Exception:
            continue

    end_time = time.time()
    print("Waktu yang dibutuhkan untuk brute force:", end_time -
start_time, "detik")

# Contoh penggunaan
public_key, private_key = generate_rsa_keypair()

ip_address = "114.122.5.15"
```

```
location_data = get_location_data(ip_address)
if location_data:
    encrypted_data = encrypt_data(location_data, public_key)
    decrypted_data = decrypt_data(encrypted_data, private_key)

    print("Data lokasi asli:", location_data)
    print("Data lokasi terenkripsi:", encrypted_data)
    print("Data lokasi didekripsi:", decrypted_data)

    # Simulasi serangan brute force
    brute_force_rsa(encrypted_data, public_key)
else:
    print("Gagal mendapatkan data lokasi.")
```

Maka hasil dari sourcode diatas, simulasi dari serangan brute force yaitu sebagai berikut:



Gambar 4.6 Proses Hasil Simulasi Serangan Brute Force

Hasil pengujian dalam mensimulasikan serangan brute force menunjukkan bahwa enkripsi RSA dengan panjang kunci 2048-bit cukup kuat untuk menahan serangan brute force, karena waktu yang dibutuhkan untuk menebak kunci sangat tidak realitis.

BAB V

PENUTUP

5.1. Kesimpulan

Dalam peneltian ini dapat memberikan beberapa poin-poin kesimpulan dalam implementasi enkripsi dan deskripsi berdasarkan alamat IP yaitu sebagai berikut:

1. Hasil dari penelitian ini menunjukkan bahwa algoritma RSA dapat di implementasikan dengan baik untuk dapat mengenkripsi data berdasarkan alamat IP yang diperoleh. Proses enkripsi menggunakan kunci public RSA yang mampu mengubah data menjadi ciphertext yang tidak dapat dibaca tanpa kunci deskripsi yang tepat.
2. Hasil penelitian ini membuktikan bahwa algoritma RSA dapat memastikan data geolokasi terenkripsi dengan baik, sehingga hanya dapat diakses oleh pihak yang memiliki kunci deskripsi yang sah. Dengan menggunakan SQLite dalam mengamankan data yang terenkripsi tidak dapat diakses oleh pihak yang tidak berwenang sehingga dapat meningkatkan keamanan data.
3. Implemenatasi Algoritma RSA menggunakan python untuk mengenkripsi dan mendeskripsi data berdasarkan alamat IP telah berhasil dilakukan. Bahasa pemograman python menyediakan berbagai pustaka dan modul yang dapat memudahkan implementasi kriptografi RSA. Pengujian yang dilakukan dalam penelitian ini dengan menggunakan bahasa pemograman python dapat menunjukkan bahwa python adalah alat yang efektif dan efisien.

5.2. Saran

Berdasarkan hasil dari implementasi RSA Cryptography algoritma dalam enkripsi data untuk manipulasi lokasi berdasarkan alamat IP maka penulis dapat memberikan beberapa saran yaitu sebagai berikut:

1. Perlu dilakukan penelitian lebih lanjut untuk enkripsi dan deskripsi berdasarkan alamat IP dengan menggunakan Cryptography yang lainnya.
2. Untuk penelitian selanjutnya untuk dapat meningkatkan keamanan, algoritma RSA dapat digunakan dalam kombinasi dengan algoritma lainnya.
3. Untuk penelitian selanjutnya diharapkan untuk dapat melalukan enkripsi dan deskripsi berdasarkan alamat IP pada implementasinya dapat diharapkan untuk menampilkan hasil enkripsi dan deskripsi dalam bentuk aplikasi maupun web.

DAFTAR PUSTAKA

Ade, B. (2022). Rancang Bangun Sistem Absensi Berbasis Face Id di Bank Mandiri Sungai Rumbai dengan Bahasa Pemograman Python. *Journal of Vocational Education and Information Technology (JVEIT)*, *3*(2), 65–70. https://doi.org/10.56667/jveit.v3i2.715

Andika, S. (2021). Implementasi Algoritma Freivlds Untuk Pembangkitan Kunci AlgoritmaRSA Pada Pengamanan Data Video. *Pelita Informatika : Informasi dan Informatika*, *10*(2), 70–77.

Ardhiansyah, M., Noris, S., & Andrianto, R. (2020). *Modul Jaringan Komputer Universitas Pamulang* (Nomor 1).

Chafid, N., & Soffiana, H. (2022). Impelementasi Algoritma Kriptografi Klasik Caesar Untuk Rancang Bangun Aplikasi E-Voting Berbasis Web (Studi Kasus : Sman 10 Tangerang). *Jurnal Ilmiah Sains dan Teknologi*, *6*(2), 133–145. https://doi.org/10.47080/saintek.v6i2.2249

Fauzi, R. (2023). Implementasi Algoritma Kriptografi Elgamal Untuk Pesan Rahasia Berbasis Web Di Markas Pmi Kota Tangerang. *50 |Jurnal Ilmu Komputer JIK*, *VI*(03), 50–54.

Feraldi, R., Khairuna, A., Hasan, M. A., Rezky, R., & Ramadhan, H. (2021). Kombinasi Algoritma Kriptografi Caesar Cipher Dan Permutation Cipher Untuk Pesan Teks Menggunakan Python. *Riau Journal of Computer Science*, *7*(1), 76–86.

Harun Alfirdaus, M., Tahir, M., Enno Dewanti, N., Ardianto, R., Nur Azurah, N., Firman Cahyono, N., & Informatika, P. (2023). Perancangan Aplikasi Enkripsi Deskripsi Mengunakan Metode Caesar Chiper Berbasis Web. *Jtmei)*, *2*(2), 64–76.

Ii, B. A. B., & Teori, L. (2016). *android yang diberi nama Gerbang Otomatis.apk. Dihubungkan dengan*. *2014*, 7–16.

Ilmiah, J., & Indonesia, M. (2023). *Mutiara*. *1*(1), 204–214.

Rezki, R., & Siahaan, R. F. (2021). Rancang Bangun Sistem Keamanan Data Digital dengan Metode RSA Berbasis Dekstop. *Jurnal Mahajana Informasi*, *6*(2), 32–40.

Rumetna, M. S. (2021). Kombinasi Gnu Privacy Guard Dan Hamming Distance Untuk Keamanan Email Serta Jalur Sertifikasi Combination of Gnu Privacy Guard and Hamming Distance for Email Security and Certification Paths. *Elektro Luceat [November]*, *7*(2), 151–160.

Studi, P., Informatika, T., Sains, F., Teknologi, D. A. N., Islam, U., & Syarif, N. (2021). *Evaluasi Kinerja Routing Protocol Ripng Dan Ospfv3 Pada Ipv6 Menggunakan Protocol Fhrp (Hsrp Dan Glbp)*.

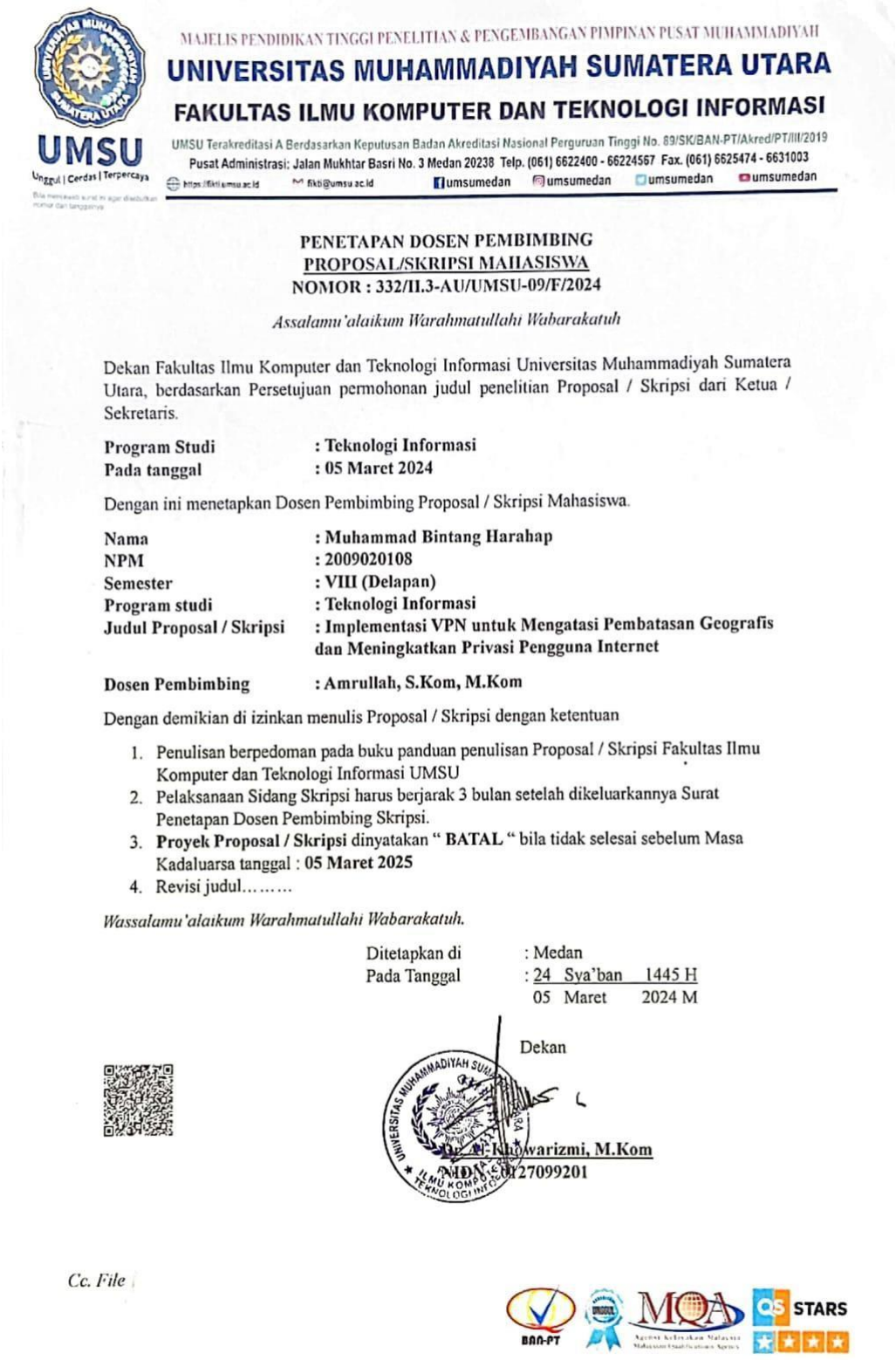
TRIANA, F. (2020). *Implementasi Caesar Cipher Cryptography Dan Least Significant Bit-2 (Lsb-2) Steganography Untuk Keamanan Data Berbasis …*. 8–42. http://eprints.polsri.ac.id/10054/

Ulfah Indriani, Ommi Alfina, N. S. (2021). *Penerapan Algoritma Rsa Dan Affine Cipher Dalam Keamanan File Ms Word*. *01*(02), 95–100. http://repository.potensi-utama.ac.id/jspui/handle/123456789/5074

Waruwu, S. H., & Hondro, R. K. (2024). *Analisis dan Implementasi Modifikasi Algoritma Kriptografi GOST Menggunakan Blum Blum Shub Generator Pada Sistem Pengamanan Login Pada Website*. *01*(03), 30–46.

LAMPIRAN-LAMPIRAN

Lampiran 1



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

UMSU Unggul | Cerdas | Terpercaya

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 332/II.3-AU/UMSU-09/F/2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi	**: Teknologi Informasi**
Pada tanggal	**: 05 Maret 2024**

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama	**: Muhammad Bintang Harahap**
NPM	**: 2009020108**
Semester	**: VIII (Delapan)**
Program studi	**: Teknologi Informasi**
Judul Proposal / Skripsi	**: Implementasi VPN untuk Mengatasi Pembatasan Geografis dan Meningkatkan Privasi Pengguna Internet**
Dosen Pembimbing	**: Amrullah, S.Kom, M.Kom**

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi** dinyatakan " **BATAL** " bila tidak selesai sebelum Masa Kadaluarsa tanggal : **05 Maret 2025**
4. Revisi judul……

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 24 Sya'ban 1445 H
05 Maret 2024 M

Dekan

Dr. Al-Khowarizmi, M.Kom
NIDN : 0127099201

Cc. File

BAN-PT MQA QS STARS

Lampiran 2

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
umsumedan umsumedan umsumedan umsumedan

SK-6

PERMOHONAN UJIAN SKRIPSI

Kepada Yth.
Bapak Dekan FIKTI UMSU
di
Medan.

Medan, 20 Juli 2024

Assalamu'alaikum wr. wb.

Dengan hormat, saya yang bertanda tangan di bawah ini Mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi UMSU:

Nama Lengkap	:	Muhammad Bintang Harahap
N P M	:	2009020108
Program Studi	:	Teknologi Informasi
Alamat Rumah	:	Jln. M. Yacub Lubis, Bandar Khalipah Gg Berkah No. 41
		Telp : 082277494274

dengan ini mengajukan permohonan mengikuti Ujian Skripsi. Bersama ini Saya lampirkan persyaratan:

1. Transkrip **Nilai Kumulatif** dari Dekan, rangkap 1;
2. Tanda **Bukti Lunas SPP** dari awal s/d akhir, rangkap 1;
3. Tanda Bukti **Lunas Biaya Ujian Skripsi**, rangkap 1;
4. Foto Copy **Ijazah Terakhir Dilegalisir**, rangkap 3;
5. **Konvensi Nilai** (bagi Mahasiswa pindahan), rangkap 1;
6. Surat **Keterangan Pindah** dari Perguruan Tinggi Asal (bagi Mahasiswa pindahan), rangkap 1;
7. Surat Keterangan **Bebas Pinjaman Buku** dari Perpustakaan UMSU, rangkap 1;
8. Foto Copy **Cover Skripsi**, rangkap 2;
9. Foto Copy **Surat Penetapan Pembimbing (SK-2)**, rangkap 2;
10. Foto Copy **KRS dan KHS**, rangkap 1; dan Melampirkan yang **Asli KHS nya**.
11. Foto Copy **KTM (Kartu Tanda Mahasiswa)**, rangkap 1;
12. Foto Copy Sertifikat Lulus **Ujian KOMPRI**, rangkap 2
13. Foto Copy Sertifikat Lulus **Ujian Kompetensi PUSKIBI**, rangkap 2
14. Foto Copy Sertifikat Lulus **Ujian Kompetensi TOEFL**, rangkap 2
15. Foto Copy Sertifikat Lulus **Ujian Kompetensi PROGRAM STUDI**, rangkap 2
16. Permohonan **Ujian Skripsi**, rangkap 1.
17. **Pas Photo Terbaru** Hitam Putih Ukuran **3 x 4 cm = 5 lembar** dan **4 x 6 = 8 lembar.**
18. **Skripsi yang telah Disahkan** lengkap diperbanyak = **3** eksemplar dan dijilid.
19. Fotocopy **KTP** yang diperbesar **2 x lipat = 1 lembar**
20. Fotocopy **Kartu Keluarga (KK)** = 1 lembar
21. Map **Warna Biru** = 2 buah

Demikianlah permohonan Saya, untuk pengurusan selanjutnya. Atas perhatian Bapak Saya ucapkan terima kasih, *Wassalam.*

Pemohon,

(MUHAMMAD BINTANG HRP)

Disetujui oleh ;

Medan,20 Juli 2024
Dekan,

(Dr.Al-Khowarizmi, S.Kom., M.Kom)
NIDN: 0127099201

Medan,20 Juli 2024
a n Rektor,
Wakil Rektor - 1

(Prof.Dr.H. Muhammad Arifin, SH.,M.Hum)
NIDN: 0013015702

Lampiran 3

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
umsumedan umsumedan umsumedan umsumedan

SK-7

SURAT PERNYATAAN

Bismillahirrohmanirrohiem.

Yang bertanda tangan di bawah ini, mahasiswa Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara :

Nama Lengkap	:	Muhammad Bintang Harahap
Tempat, Tgl. Lahir	:	Bandar Setia, 21 Desember 2001
Agama	:	Islam
Status Perkawinan	:	Belum Kawin
N P M	:	2009020108
Alamat Rumah	:	Jln. M. Yacub Lubis, Bandar Khalipah Gg.Berkah No. 41
		Telp/HP. 082277494274
Pekerjaan/Instansi	:	-
Alamat Kantor	:	-
		Telp/HP. -

melalui surat permohonan tertanggal 20 Juli 2024 telah mengajukan permohonan menempuh Ujian Skripsi. Untuk ujian skripsi yang akan saya tempuh, menyatakan dengan sesungguhnya :

1. Bahwa saya dalam keadaan sehat jasmani dan rohani.
2. Bahwa saya siap secara optimal dan berada dalam kondisi baik untuk memberikan jawaban atas pertanyaan dari Penguji.
3. Bahwa saya bersedia menerima keputusan yang ditetapkan oleh Panitia Penguji Skripsi dengan ikhlas tanpa mengadakan gugatan apapun juga.
4. Saya menyadari bahwa keputusan Panitia Penguji ini bersifat mutlak dan tidak dapat diganggu gugat.

Demikianlah surat pernyataan ini saya buat dengan kesadaran tanpa paksaan atau tekanan dalam bentuk apa pun dan dari siapa pun, untuk dipergunakan bilamana dipandang perlu. Semoga Allah SWT. meridhoi saya. Amin.-

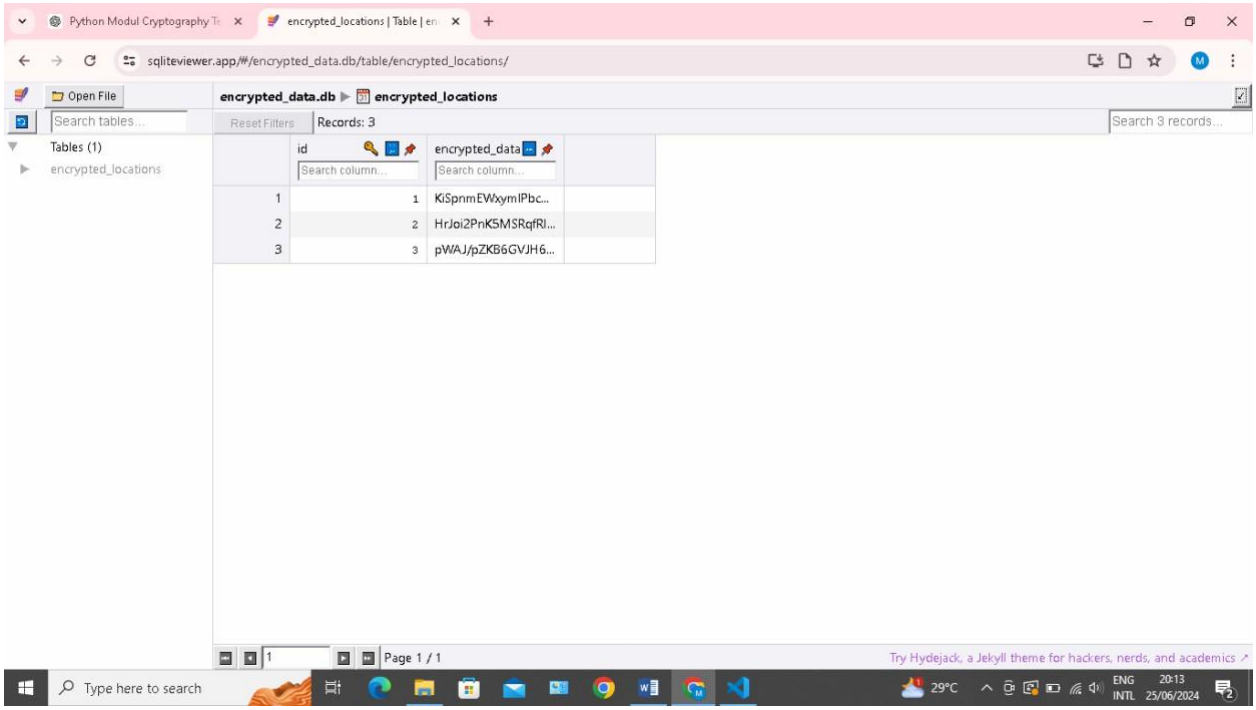
Saya yang menyatakan,

METERAI TEMPEL

MUHAMMAD BINTANG HRP

SK-8

Lampiran 4


Python Modul Cryptography
encrypted_locations | Table |
sqliteviewer.app/#/encrypted_data.db/table/encrypted_locations/
Open File
encrypted_data.db
encrypted_locations
Search tables...
Reset Filters
Records: 3
Search 3 records...
Tables (1)
encrypted_locations
id
encrypted_data
Search column...
Search column...
1 1 KiSpnmEWxymIPbc...
2 2 HrJoi2PnK5MSRqfRl...
3 3 pWAJ/pZKB6GVJH6...
1
Page 1 / 1
Try Hydejack, a Jekyll theme for hackers, nerds, and academics
Type here to search
29°C
ENG
INTL
20:13
25/06/2024