Jurnal Masyarakat Informatika, 15(1), 2024
DOI:10.14710/jmasif.15.2.66317
ISSN: 2777-0648

Kriptografi Homomorfik dalam Anonimisasi Data untuk Pengolahan Data pada Sistem *E-Voting*

Muhammad Naufal Zhafran Soleh*, Asep Id Hadiana, Fatan Kasyidi
Jurusan Informatika, Universitas Jenderal Ahmad Yani, Indonesia
* *Corresponding author: muhammadnaufalzs20@if.unjani.ac.id*

Abstrak

Dalam era kemajuan teknologi informasi, menjaga keamanan dan privasi data merupakan suatu tantangan yang sangat penting. Penelitian ini bertujuan untuk membuktikan teknik keamanan Kriptografi dengan metode Homomorfik yang dapat mengolah data pada saat data terenkripsi menyoroti kekurangan dalam implementasi enkripsi standar yang hanya mengenkripsikan data saja. Metode penelitian ini meliputi analisis terhadap Kriptografi Homomorfik, dengan membangun sistem e-voting untuk menerapkan metode Homomorfik dengan algoritma Pailier yang dapat mengolah data vote pada saat data terenkripsi, pembuatan kunci yang dapat Enkripsi data vote dan Dekripsi data vote. Hasil dari penelitian ini adalah penerapan Kriptografi dengan metode Homomorfik dalam sistem website e-voting mendapatkan keberhasilan pengolahan data pada saat data terenkripsi kemudian dari hasil penyerangan pengujian keamanan menggunakan Chipher text only attack tidak dapat ditemukan data plaintext dan pengujian performa waktu mendapatkan hasil 0.010 detik untuk proses enkripsi dekripsi, serta pengujian Shannon entropy mendapatkan hasil 3.20 bits dari 1 data vote terenkripsi.

Kata kunci : *kriptografi, homomorfik, pailiier, enkrispi, dekripsi, e-voting*

Abstract

In today's age of technological advancement, protecting data security and privacy is a significant challenge. This study seeks to prove the effectiveness of Homomorphic Cryptography techniques, which allow data to be processed while remaining encrypted, addressing the limitations of standard encryption methods that only secure data but do not enable processing of encrypted information. The research approach involves analyzing Homomorphic Cryptography by designing an e-voting system that utilizes the Paillier algorithm, enabling the processing of vote data while it is encrypted, along with generating keys for encrypting and decrypting the votes. The results of this study indicate that implementing Homomorphic Cryptography within a web-based e-voting platform successfully facilitates data processing in an encrypted state. Additionally, security testing via a Ciphertext-only attack did not reveal any plaintext information, while performance testing recorded an encryption-decryption time of 0.010 seconds. Shannon entropy analysis also yielded a result of 3.20 bits for a single encrypted vote.

Keywords : *cryptography, homomorphic, pailier, encryption, decryption, e-voting,*

1 Pendahuluan

Kemajuan pesat dalam era teknologi informasi digital telah membawa dampak positif yang signifikan pada berbagai aspek kehidupan, terutama dalam hal pengelolaan data terkait keamanan dan privasi. Seiring dengan peningkatan volume dan ragam data yang terus berkembang, tantangan dalam

Received: 28 Agustus 2024 ; Revised: 28 November 2024; Accepted: 29 November 2024; Published: 30 November 2024

melindungi informasi pribadi semakin besar. Data tersebut tersebar di berbagai platform dan layanan, sehingga keamanan data menjadi lebih rentan terhadap serangan siber dan pelanggaran privasi, yang kini menjadi ancaman serius. [1]

Dalam era digital yang semakin berkembang, pengelolaan data sensitif seperti catatan medis elektronik, pemungutan suara, transaksi keuangan, dan data pribadi membutuhkan solusi keamanan yang lebih inovatif. Salah satu metode yang dapat diterapkan adalah kriptografi homomorfik, yang memungkinkan komputasi dilakukan langsung pada data terenkripsi tanpa harus melakukan dekripsi terlebih dahulu. Pendekatan ini menawarkan peluang untuk menjaga anonimitas data dengan aman serta memungkinkan pengolahan data terenkripsi secara efisien, sekaligus memastikan kepatuhan terhadap regulasi privasi seperti General Data Protection Regulation (GDPR) [2]

Penelitian terdahulu menyoroti upaya menjaga kerahasiaan data melalui penerapan dan desain sistem keamanan file yang menggunakan metode enkripsi AES (Advanced Encryption Standard). Untuk melindungi informasi yang tersimpan dalam blok 128-bit, digunakan teknik enkripsi dan dekripsi dengan kunci berukuran 128, 192, dan 256 bit. Metode ini terdiri dari empat tahap transformasi byte, yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey sebagai bagian dari proses enkripsi. Hasil penelitian ini adalah implementasi dan pengujian sistem yang melibatkan proses enkripsi-dekripsi untuk mengonversi berbagai jenis file, seperti teks, dokumen, Gambar, dan video, menjadi ciphertext (enkripsi) dan kemudian mengembalikannya ke bentuk asli (dekripsi). Pengujian ini juga mencakup serangan XLS Attack, yaitu metode serangan yang diklaim dapat memecahkan AES lebih cepat dibandingkan pencarian menyeluruh, dengan menggunakan analisis subsistem internal cipher untuk mereduksi persamaan kuadrat. Namun, kelemahan dari metode AES ini adalah ketidakmampuannya untuk melakukan operasi langsung pada data terenkripsi jadi proses untuk melakukan operasi pada data terenkripsi harus melakukan dekripsi terlebih dahulu. [3]

Kriptografi homomorfik menawarkan manfaat signifikan dengan memungkinkan operasi komputasi dilakukan langsung pada data yang terenkripsi tanpa perlu mendekripsinya terlebih dahulu. Ini membuka peluang baru untuk memproses data dengan tetap mempertahankan tingkat keamanan yang tinggi. Kriptografi homomorfik memiliki peran penting dalam situasi di mana privasi data menjadi prioritas utama, menawarkan solusi efektif dalam skenario yang memerlukan perlindungan privasi yang ketat, penelitian ini membangun sistem e-voting untuk membuktikan penerapan kriptografi dengan menggunakan metode homomorfik. Metode ini dianggap sebagai solusi atas kelemahan metode AES (Advanced Encryption Standard), DES (Data Encryption Standard), serta RSA (Rivest-Shamir-Adleman), yang memerlukan dekripsi sebelum melakukan operasi pada data [4].

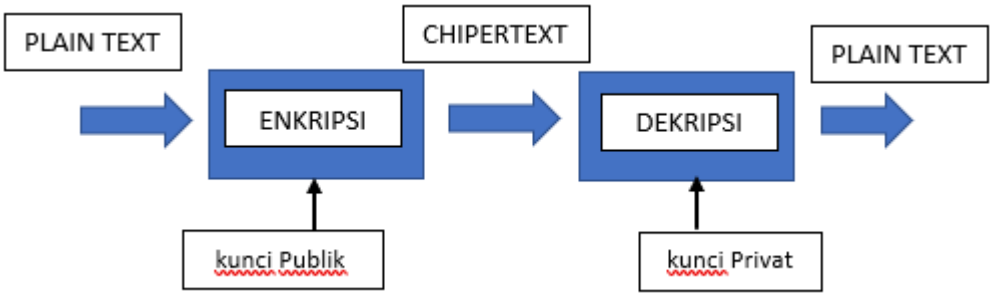
Metode enkripsi AES (Advanced Encryption Standard), DES (Data Encryption Standard), serta RSA (Rivest-Shamir-Adleman) memiliki kelemahan yang signifikan, yaitu data harus melalui proses dekripsi sebelum dapat diolah. Proses ini berpotensi menimbulkan risiko terhadap privasi data, terutama jika data yang telah didekripsi dapat diakses atau dimanipulasi oleh pihak yang tidak bertanggung jawab. Sebagai solusi atas keterbatasan ini, kriptografi homomorfik hadir sebagai teknologi inovatif. Dengan teknologi ini, operasi komputasi dapat dilakukan langsung pada data yang masih terenkripsi tanpa perlu mendekripsinya terlebih dahulu. Dalam sistem e-voting, pendekatan ini menawarkan keuntungan besar karena data suara tetap dalam keadaan terenkripsi selama proses penghitungan berlangsung, sehingga keamanan data menjadi lebih terjamin. Selain itu, kriptografi homomorfik juga berperan dalam anonimisasi data pemilih, memastikan privasi individu tetap terjaga baik selama maupun setelah proses pemungutan suara selesai.

Penelitian ini menawarkan pendekatan baru dengan mengadopsi kriptografi homomorfik Paillier sebagai solusi utama untuk mengatasi kelemahan tersebut dalam sistem e-voting. Teknologi ini memungkinkan penghitungan suara secara langsung pada data terenkripsi tanpa memerlukan dekripsi, serta mendukung proses anonimisasi data secara aman. Kebaruan yang diusung dalam penelitian ini adalah integrasi antara anonimisasi data dan pengolahan data terenkripsi, dengan kemampuan memproses data tanpa dekripsi, pendekatan ini menjawab tantangan utama dalam keamanan e-voting modern.dengan pendekatan ini, penelitian memberikan kontribusi penting dalam meningkatkan keamanan dan privasi sistem *e-voting* modern.

2 Penelitian Terkait

2.1 Kriptografi

Kriptografi merupakan teknik yang digunakan untuk melindungi keamanan informasi dengan cara menyandikan data. Proses utama dalam kriptografi meliputi enkripsi dan dekripsi [5]Informasi yang akan dienkripsi, yang dikenal sebagai plaintext, dapat dibaca dan dipahami dengan mudah oleh siapa saja. Algoritma yang digunakan dalam enkripsi dan dekripsi memerlukan kunci tertentu. Pesan plaintext yang telah dienkripsi (atau dikodekan) disebut ciphertext, sementara dekripsi adalah proses mengembalikan ciphertext ke bentuk plaintext aslinya. Penelitian ini memanfaatkan kriptografi dengan tujuan utama untuk menjaga kerahasiaan data dan mencegah akses yang tidak sah terhadap informasi sensitif. Secara sederhana, proses dari enkripsi ke dekripsi yang ditunjukkan pada Gambar 1



Gambar 1 Konsep Kriptografi

2.2 Anonimisasi Data

Strategi anonimisasi data menjadi krusial untuk melindungi identitas individu yang terkait dengan informasi yang dikumpulkan. Anonimisasi data adalah teknik yang digunakan untuk menjaga privasi dengan menghapus atau mengenkripsi informasi [6]. Proses ini melibatkan penghapusan identitas pribadi dari dataset, sehingga individu yang datanya tercakup tidak dapat lagi diidentifikasi. Kriptografi homomorfik menawarkan solusi penting untuk mengatasi tantangan ini, memungkinkan pengguna untuk memanfaatkan layanan cloud tanpa kehilangan kendali atas privasi data pribadi mereka [7].

Anonimisasi melibatkan penghapusan data pribadi seperti nama, alamat, atau informasi pengenal lainnya dari dataset. Tujuan utamanya adalah agar data tetap berguna untuk penelitian tanpa mengorbankan perlindungan identitas individu. Keunggulan dari teknik anonimisasi adalah kemampuannya untuk menghasilkan data yang dapat dianalisis tanpa merusak privasi individu.

Dengan cara ini, risiko penyalahgunaan atau akses tidak sah terhadap data dapat dikurangi secara signifikan.

2.3 Pembentukan Kunci

Teknologi kriptografi kunci publik, atau yang juga dikenal sebagai kriptografi asimetris, merujuk pada sebuah sistem kriptografi yang mengadopsi pasangan kunci, yakni kunci publik serta kunci privat. Kunci publik, sebagai bagian dari pasangan tersebut, dirancang untuk dapat didistribusikan secara terbuka. Sebaliknya, kunci privat merupakan rahasia yang hanya dimiliki oleh pihak yang berwenang. Dalam proses enkripsi, salah satu kunci digunakan, sementara kunci lainnya dipergunakan untuk dekripsi. Keunggulan utama dari kriptografi kunci publik adalah eliminasi kebutuhan akan mekanisme pengiriman kunci rahasia. Terdapat 2 kunci yaitu kunci publik dan kunci privat yang dijelaskan oleh Rinaldi Munir [8]:

1. Menentukan 2 buah bilangan prima p dan q dapat dilakuan secara acak, yang dapat memenuhi :

$$fpb\big(pq,(p-1)(q-1)\big)=1 \quad (1)$$

Dimana *fbp* adalah faktor persekutuan terbesar

2. Menghitung :

$$n=pq \text{ dan } \lambda=kpk(p-1,q-1) \quad (2)$$

Dimana *kpk* adalah faktor persekutuan terkecil

3. Memilih sembarang bilangan bulat *g*, dengan $g<n^2$
4. Menghitung bahwa *n* habis dibagi *g*, dengan persamaan berikut :

$$\mu=(L\big(g^{\lambda}mod\ n^2\big))^{-1}\ mod\ n \quad (3)$$

Fungsi L adalah :

$$L(\mathrm{x})\ =\ \frac{x-1}{n} \quad (4)$$

5. Mendapatkan hasil berupa kunci publik (g,n) dan kunci privat (λ,μ)

2.4 Enkripsi

Secara umum, proses enkripsi melibatkan pengubahan informasi menjadi kode rahasia melalui langkah-langkah teknis. Data yang dikirim, diterima, atau disimpan akan diubah menjadi bentuk yang tidak dapat dibaca. Biasanya, algoritma digunakan untuk mengacak data tersebut, dan penerima dapat mengembalikannya ke bentuk semula menggunakan kunci dekripsi. Informasi yang telah dipulihkan dan masih dalam format tidak terenkripsi disebut *plaintext*, sedangkan informasi yang telah diubah menjadi format terenkripsi dikenal sebagai *ciphertext* [9]. Proses enkripsi dengan Homomorfik Encryption sebagai berikut :

1. Terdapat bilangan m yang menunjukan m adalah data pesan yang akan di enkripsi, yaitu dengan syarat $0\le m<n$
2. Kemudian pilih bilangan bulat secara acak *r* dimana $0\le r<n$ dan PBB $(r,n)=1$
3. Selanjutnya menghitung cipherteks dari m dengan persamaan sebagai berikut :

$$c = g^m . r^n \; mod \; n^2 \quad (5)$$

Dimana c yaitu residu ke -n dalam modulus n^2 dengan simbol $[c]_g$.

2.5 Dekripsi

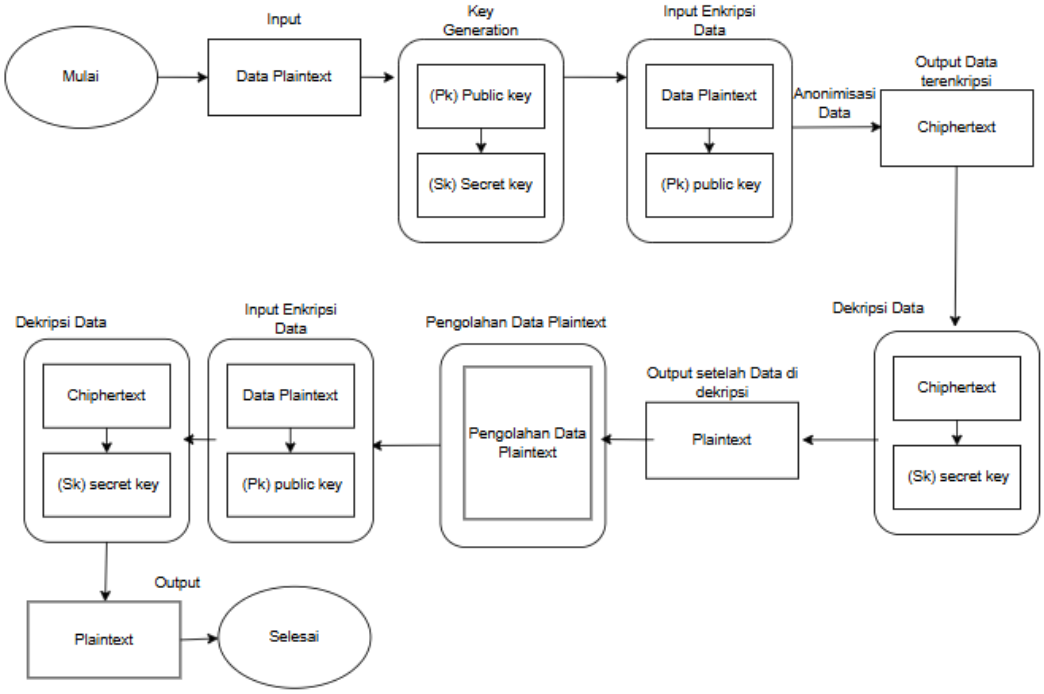
Dekripsi adalah kebalikan dari enkripsi, di mana data atau pesan yang sebelumnya diubah menjadi bentuk terenkripsi (ciphertext) dikembalikan ke format aslinya yang dapat dibaca oleh manusia (plaintext). Proses dekripsi ini memerlukan kunci kriptografi yang cocok dengan data tersebut untuk mengembalikan informasi ke kondisi aslinya [10]. Algoritma untuk melakukan proses dekripsi data pada homomorfik yaitu :

1. *c* adalah chiphertext yang akan di dekripsikan, dimana c yaitu ∈ $\mathbb{Z}_{n^2}^*$
2. Menghitung plaintext dari *c* tersebut dengan persamaan :

$$m = L\left(c^{\lambda} mod \; n^2\right).\mu \; mod \; n \quad (6)$$

2.6 Kriptografi AES (Advanced Encryption Standard) dan RSA (Rivest-Shamir-Adleman)

Konsep Proses Kriptografi Asimetris Rivest-Shamir-Adleman (RSA) suara individu harus didekripsi terlebih dahulu untuk diproses. Setelah penghitungan selesai, data sering kali perlu dienkripsi kembali, menjadikan proses RSA lebih kompleks dan rawan terhadap kebocoran informasi. Pada kriptografi homomorfik, data pemilih (suara) dienkripsi menggunakan kunci publik dan disimpan dalam bentuk terenkripsi (ciphertext). Operasi matematika, seperti penjumlahan suara, dapat dilakukan langsung pada ciphertext tanpa perlu mendekripsinya. Dekripsi hanya dilakukan sekali, yaitu setelah seluruh perhitungan selesai, dengan menggunakan kunci privat untuk mendapatkan hasil akhirnya pada Gambar 2.



Gambar 2 Proses Kriptografi RSA

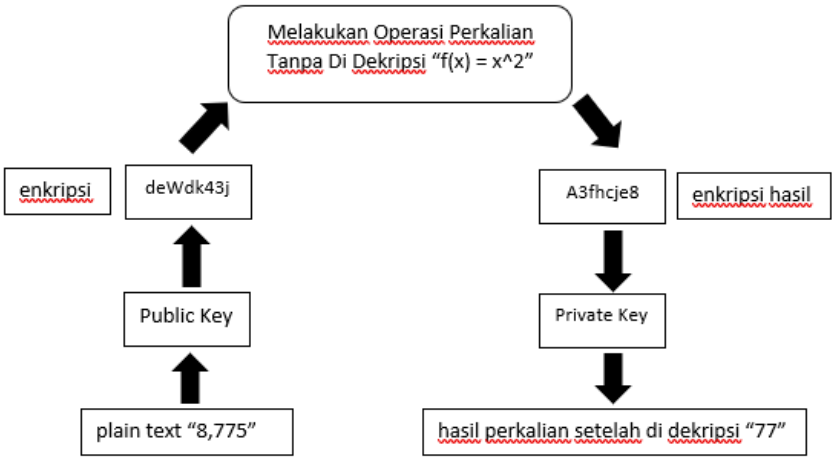
Meskipun metode enkripsi tradisional seperti AES (Advanced Encryption Standard) dan RSA (Rivest-Shamir-Adleman),telah banyak digunakan dalam sistem *e-voting*, terdapat beberapa

keterbatasan signifikan. Salah satu kelemahannya adalah ketidakmampuan untuk melakukan operasi komputasi langsung pada data yang telah dienkripsi. Proses seperti penjumlahan atau penghitungan suara memerlukan tahap dekripsi terlebih dahulu, yang dapat membuka potensi risiko keamanan jika data terekspos selama proses tersebut. Selain itu, metode enkripsi konvensional kurang mendukung otomatisasi anonimisasi data, sehingga identitas pemilih dapat terancam jika pengelolaan data tidak dilakukan dengan cermat

2.7 Homomorfik

Pemrosesan data homomorfik menjadi komponen penting dalam menjaga privasi dan keamanan informasi. Teknik ini memungkinkan komputasi dilakukan langsung pada data terenkripsi tanpa perlu mendekripsinya terlebih dahulu. Dengan kata lain, data tetap dalam bentuk terenkripsi (ciphertext) selama proses perhitungan, dan hasil akhirnya hanya dapat dimengerti setelah melalui proses dekripsi. Pemrosesan data homomorfik menggunakan operasi matematika kriptografi yang dirancang untuk memungkinkan penghitungan pada data terenkripsi, sembari memastikan bahwa keamanan dan integritas data tetap terlindungi. Algoritma yang digunakan dibuat agar operasi dapat dilakukan tanpa mengungkapkan data asli.

Diagram enkripsi homomorfik pada Gambar 3 menunjukkan bahwa proses ini memungkinkan penghitungan dilakukan pada data terenkripsi tanpa dekripsi, sehingga memungkinkan pemrosesan data yang aman tanpa mengorbankan keamanan [11].



Gambar 3 Konsep Homomorfik

Proses implementasi sifat Homomorfik dengan Algoritma Pailier terdapat dua buah data plaintext adalah m_1 dan m_2 yang hasil enkripsi yaitu $c_1$dan c_2 dalam persamaan berikut :

$$E(m_1, r_1) = c_1 = g^m . r_1^n \; mod \; n^2 \tag{7}$$

$$E(m_2, r_2) = c_2 = g^m . r_2^n \; mod \; n^2 \tag{8}$$

kemudian dapat mengkalikan ciphertext $c_1$dan c_2 dalam persamaan berikut :

$$E(m_1, r_1) \, . E(m_2, r_2) \; = c_1 . c_2 = g^{m1} . r_1^n . g^{m2} . r_2^n \; mod \; n^2$$

$$= g^{m1+m2} (r_1 . r_2)^2 mod \; n^2 \tag{9}$$

selanjutnya dapat ditunjukan bahwa hasil dekripsi terhadap $c_1.\ c_2$ yaitu sama dengan penjumlahan dari kedua plaintextnya , yaitu dalam persamaan berikut :

$$D(c_1, c_2) = m_1 + m_2 \ mod\ n \tag{10}$$

2.8 Partial Homomorfik Encryption

Jenis enkripsi homomorfik [12]:

1. Enkripsi Homomorfik Sebagian (Partially Homomorphic Encryption): Jenis ini memungkinkan hanya satu jenis operasi aritmetika, seperti penjumlahan atau perkalian, pada ciphertext. Enkripsi homomorfik sebagian bersifat aditif atau multiplikatif, yang berarti dua ciphertext hanya bisa dijumlahkan atau dikalikan, tetapi tidak keduanya.
2. Enkripsi Homomorfik Penuh (Fully Homomorphic Encryption): Pada jenis ini, baik operasi penjumlahan maupun perkalian dapat dilakukan pada ciphertext, sehingga bersifat aditif dan multiplikatif. Saat dua ciphertext dijumlahkan atau dikalikan, hasil dekripsinya akan sama dengan hasil penjumlahan atau perkalian plaintext asli.

Dari 2 jenis tersebut Partially Homomorphic Encryption mempunyai Algoritma Paillier yaitu contoh algoritma enkripsi homomorfik yang memiliki sifat aditif, di mana hasil dekripsi dari perkalian dua cipherteks setara dengan penjumlahan dari kedua plainteks tersebut. Pernyataan tersebut dapat dibuktikan dengan contoh studi kasus yaitu sebagai berikut :

1. Terdapat plaintext m_1= 42 dengan r_1= 23 hasil ciphertext yaitu c_1= 4624 dan m_2= 29 dengan r_2 = 30 hasil ciphertext nya c_2= 1539
2. Kemudian sifat homomorfik dapat dilakukan disini yaitu dapat melakukan perkalian dari kedua ciphertext yaitu $c_1 . c_2$ = 4624 . 1539 = 7116336 mod 5929 = 1536
3. Selanjutnya melakukan dekripsi dari $c_1 . c_2$ adalah :

$$\begin{aligned} D(c_1.c_2) &= L\big((c_1 . c_2)^{\lambda}\ mod\ n^2\big).\mu\ mod\ n \\ &= L(1536^{30}\ mod\ 77^2).74\ mod\ 77 \\ &= L(1536^{30}\ mod\ 5929).74\ mod\ 77 \\ &= L(155).74\ mod\ 77 \\ &= (155-1)/77.74\ mod\ 77 \\ &= 2\ .74\ mod\ 77 \\ &= 148\ \ mod\ 77 \\ &= 71 \end{aligned} \tag{11}$$

4. Hasil dari dekripsi diatas yaitu sama dengan penjumlahan dua buah plaintext dalam modulus 77, yaitu $(m_1 + m_2)\ mod\ n = (42 + 29)\ mod\ 77 = 71\ mod\ 77 = 71$
5. Hasil akhir dapat dibuktikan bahwa $D(c_1.c_2) = \ (m_1 + m_2)\ mod\ n = 71$

Pada penelitian lain menyebutkan bahwa Sebuah sistem kriptografi dikenal sebagai "Enkripsi Homomorfik Parsial (PHE)" ketika menunjukkan sifat homomorfisme baik dalam bentuk perkalian atau penjumlahan, tetapi tidak keduanya sekaligus. Beberapa contoh paling terkenal dari sistem kriptografi homomorfik parsial adalah RSA dan ElGamal (keduanya memiliki homomorfisme

multiplikatif), dan Paillier (homomorfisme aditif). Adapun perbedaan dari Additive dan Multiplicative di jelaskan dalam persamaan berikut ini :

Additive Homomorphic:

$$Enc(x \oplus y) = Enc(x) \otimes Enc(y) \quad (12)$$

$$Enc(\sum_{i=1}^{1} m_i) = \prod_{i=1}^{1} Enc(m_i) \quad (13)$$

Multiplicative Homomorphic:

$$Enc(x \otimes y) = Enc(x) \otimes Enc(y) \quad (14)$$

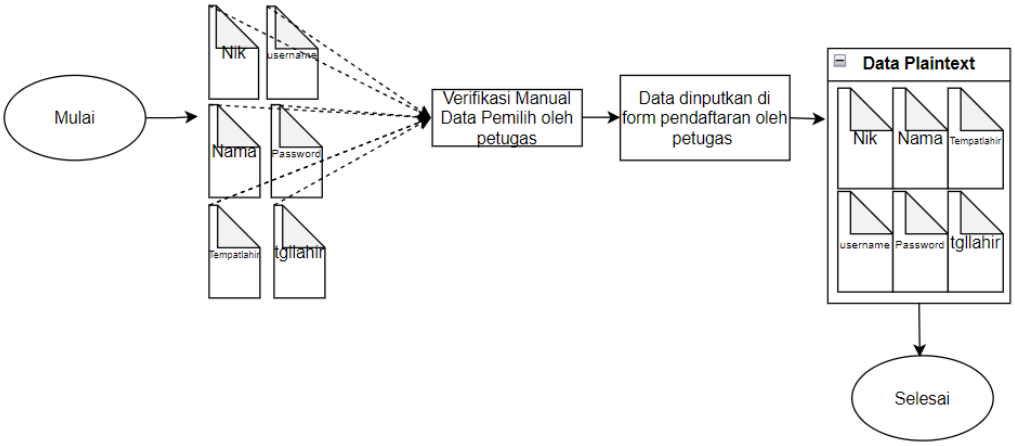
$$Enc(\sum_{i=1}^{1} m_i) = \prod_{i=1}^{1} Enc(m_i) \quad (15)$$

3 Metode Penelitian

Metode penelitian ini terbagi menjadi dua bagian, yaitu proses anonimisasi data yang bertujuan untuk menyembunyikan identitas pribadi peserta pemilih dan proses metode homomorfik yang digunakan untuk mengolah data suara saat data masih dalam bentuk terenkripsi. Dengan metode Homomorfik ini, data yang telah terenkripsi tidak perlu didekripsi terlebih dahulu sebelum diproses.

3.1 Input Data

Tahap pertama adalah melakukan input data, di mana data yang diperoleh mencakup informasi identitas penduduk yang akan berpartisipasi dalam voting. Data ini berasal dari setiap peserta pemilih dan mencakup NIK (Nomor Induk Kependudukan), Nama, Tempat lahir, dan Tanggal lahir. Selain itu, data input juga mencakup username dan password yang dibuat serta dikelola secara manual oleh petugas sebelum masuk ke sistem. Seluruh data ini kemudian digunakan oleh peserta pemilih untuk validasi login ke dalam sistem e-voting dengan status sah dan terdaftar. Tahap input data ini diGambarkan pada Gambar 4

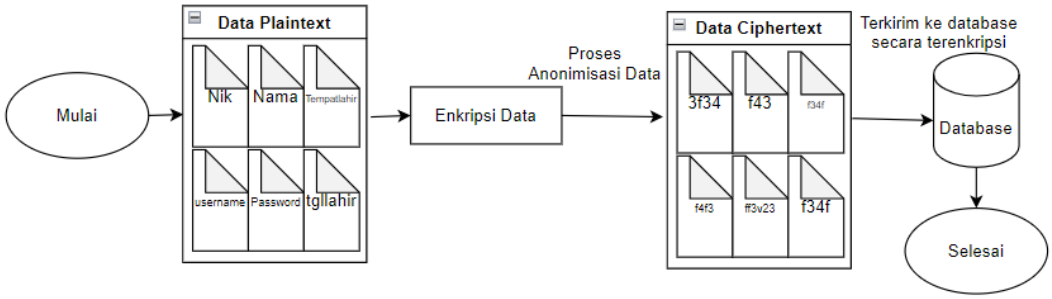


Gambar 4 Proses Input Data Anonimisasi

Proses yang dilakukan pada tahapan ini adalah menginputkan semua data diri peserta pemilihan yang nantinya akan tersimpan dahulu secara terenkrpsi ke database yang selanjutnya akan di anonimkan pada saat pemilhan suara

3.2 Anonimisasi

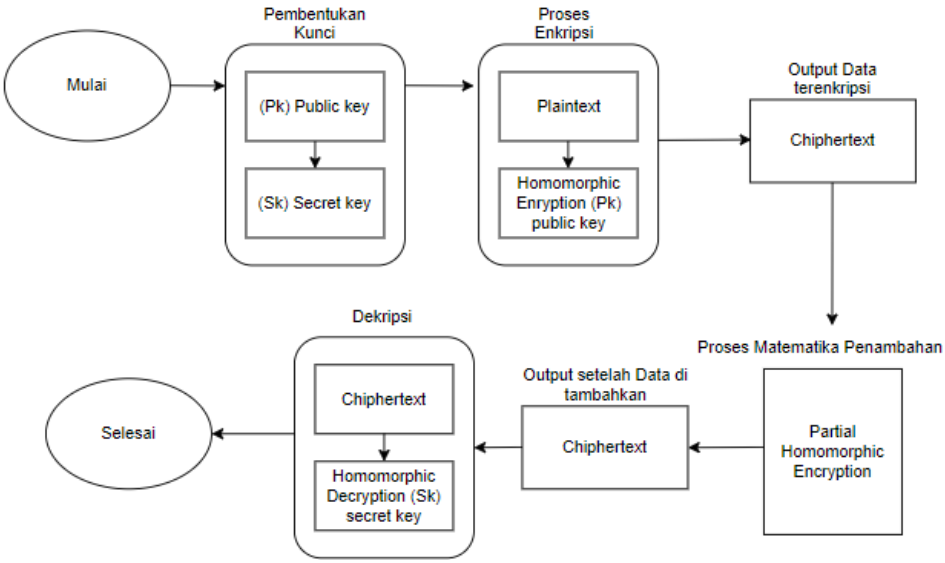
Tahapan Anonimisasi data adalah proses untuk menghilangkan identitas data pribadi peserta pemilih yang tidak dapat dibaca kembali oleh pihak manapun, akan tetapi data tersebut masih tersimpan di database dan dapat digunakan untuk melakukan verifikasi login peserta pemilih, tahapan anonimisasi data pemilih terdapat pada Gambar 5



Gambar 5 Proses Anonimisasi

3.3 Homomorfik

Tahapan ini menjelaskan terhadap proses pengelolaan suara masuk yang akan dilakukan rekapitulasi suara menggunakan proses penjumlahan terhadap implementasi homomorfik pada Gambar 6 adalah Konsep tahapan implementasi melakukan proses homomorfik.



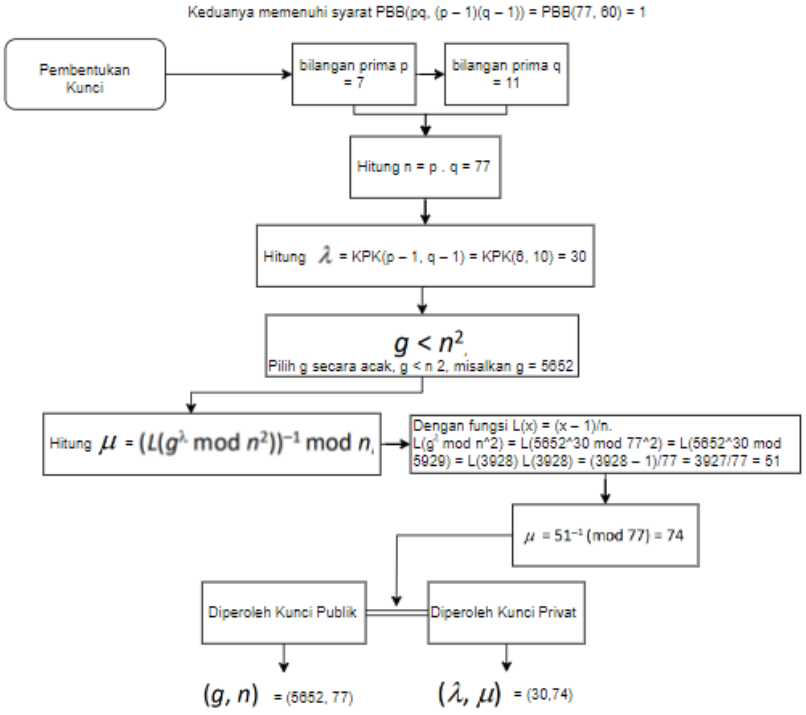
Gambar 6 Proses Kriptografi Homomorfik

Kriptografi homomorfik memungkinkan pengolahan data dilakukan langsung pada data yang telah terenkripsi tanpa memerlukan proses dekripsi. Dengan pendekatan ini, data tetap dalam bentuk terenkripsi (ciphertext) selama penghitungan, sehingga privasi dan keamanan tetap terjamin. Kriptografi homomorfik memungkinkan operasi langsung pada data terenkripsi, sedangkan pada kriptografi asimetris seperti RSA, data harus didekripsi terlebih dahulu sebelum dapat diproses. Perbedaan ini menyoroti keunggulan kriptografi homomorfik dalam memproses data tanpa dekripsi, sehingga meningkatkan efisiensi dan keamanan.

Kesesuaian untuk E-Voting : Kriptografi homomorfik sangat sesuai untuk aplikasi e-voting karena mampu menjaga anonimitas dan privasi pemilih. Suara tetap dalam bentuk terenkripsi hingga hasil akhirnya diumumkan, memastikan data tidak pernah terpapar. Sebaliknya, RSA, yang memerlukan dekripsi sebelum penghitungan, tidak dapat memberikan tingkat privasi yang sama, menjadikannya kurang cocok untuk sistem e-voting. Dengan kemampuan menjaga keamanan dan efisiensi secara bersamaan, kriptografi homomorfik adalah pilihan yang lebih ideal untuk pemilu elektronik.

3.4 Pembentukan kunci

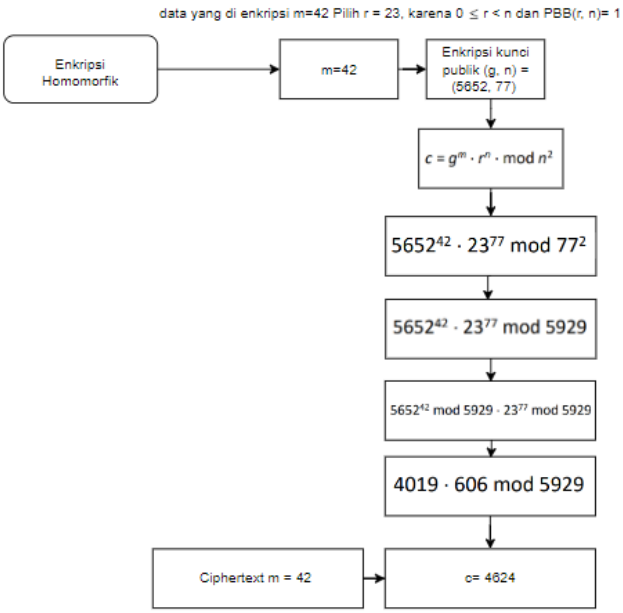
Tahapan pembentukan kunci Dalam algoritma Paillier, kunci publik berperan dalam proses enkripsi data, sementara kunci privatnya digunakan untuk melakukan dekripsi data, Enkripsi dilakukan dengan menggunakan salah satu dari kedua kunci tersebut, sedangkan dekripsi menggunakan kunci yang lain. Salah satu manfaat dari kriptografi kunci publik adalah tidak memerlukan mekanisme khusus untuk pengiriman kunci rahasia, proses pembentukan kunci dijelaskan pada Gambar 7



Gambar 7 Proses Pembentukan Kunci

3.5 Enkripsi Homomorfik

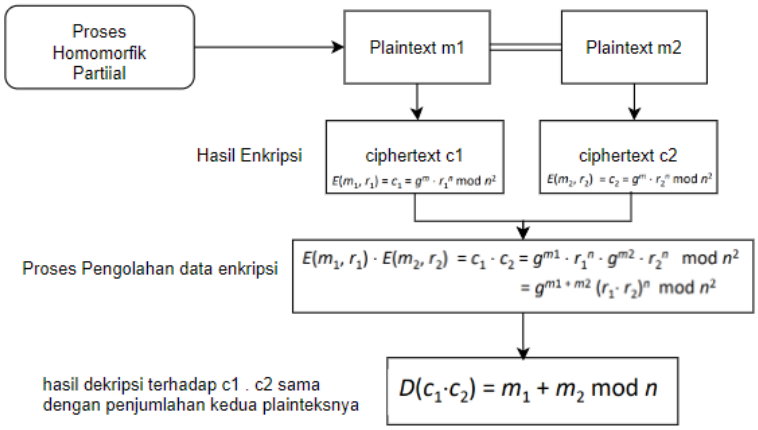
Tahapan selanjutnya yaitu enkripsi, secara sistematis proses enkripsi homomorfik mengubah satu set data menjadi set data lainnya sambil mempertahankan hubungan antara kedua set data tersebut, pada proses implementasi ini mengambil studi kasus dengan contoh terdapat plaintext m = 42 yang akan di enkripsi dengan hasil enkripsi menjadi c= 4624, proses enkripsi di jelaskan pada Gambar 8.



Gambar 8 Proses Enkripsi Homomorfik

3.6 Pengolahan Enkripsi Homomorfik

Pengolahan Enkripsi Homomorfik ini bertujuan untuk menjelaskan bagaimana data yang telah dienkripsi dapat diproses tanpa perlu didekripsi terlebih dahulu, dengan menggunakan metode partial homomorphic encryption.



Gambar 9 Proses Enkripsi Homomorfik

Proses ini menghasilkan enkripsi $m1 = 42$ dengan $r1 = 23$ yang menghasilkan ciphertext $c1 = 4624$. Misalkan nilai $m2 = 29$ dan $r2 = 30$, setelah dihitung, ciphertext-nya adalah $c2 = 1539$. Perkalian kedua ciphertext tersebut adalah. $c1 \times c2 = 4624 \times 1539 = 7116336 \mod 5929 = 1536$. Dekripsi dari c1 × c2 adalah:

$$D(c1 \times c2) = L((c1 \times c2)\text{^}\lambda \mod n\text{^}2) \times \mu \mod n \quad (16)$$

$$= L(1536\text{^}30 \mod 772) \times 74 \mod 77$$

$= L(153630 \bmod 5929) \times 74 \bmod 77$

$= L(155) \times 74 \bmod 77$

$= (155 - 1)/77 \times 74 \bmod 77$

$= 2 \times 74 \bmod 77$

$= 148 \bmod 77$

$= 71$

Hasil dekripsi tersebut setara dengan penjumlahan dua plainteks dalam modulus 77

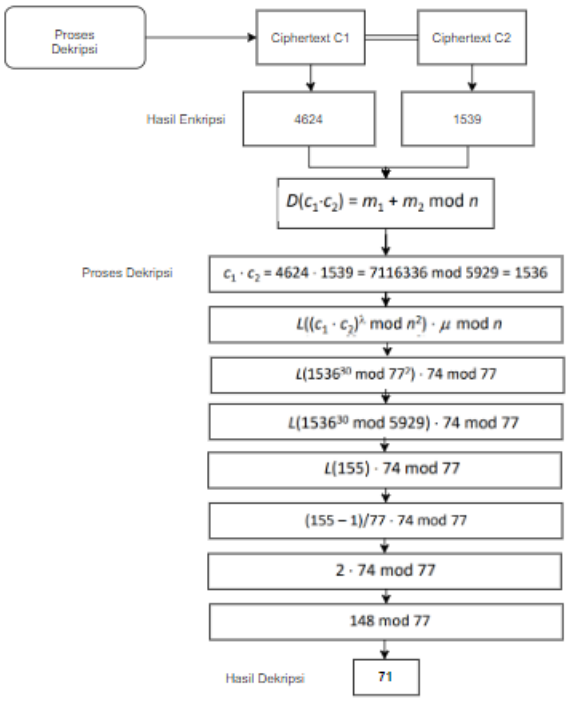
$$= (m1 + m2) \bmod n \quad (17)$$

$= (42 + 29) \bmod 77$

$= 71 \bmod 77$

$= 71.$

Dengan demikian, $D(c1 \times c2) = (m1 + m2) \bmod n = 71$.



Gambar 10 Proses Dekripsi Homomorfik

3.7 Dekripsi Homomorfik

Langkah terakhir dalam implementasi ini adalah dekripsi, di mana setelah proses pengolahan data pada saat data masih dalam keadaan terenkripsi, hasilnya kemudian didekripsi., Dekripsi dilakukan

dengan menghitung plainteks menggunakan kunci privat (λ, μ) = (30, 74) Hasil dekripsi tersebut sesuai dengan penjumlahan dua plainteks dalam modulus 77, yaitu

$$= (m1 + m2) \bmod n \quad (18)$$

$$= (42 + 29) \bmod 77$$

$$= 71 \bmod 77$$

$$= 71.$$

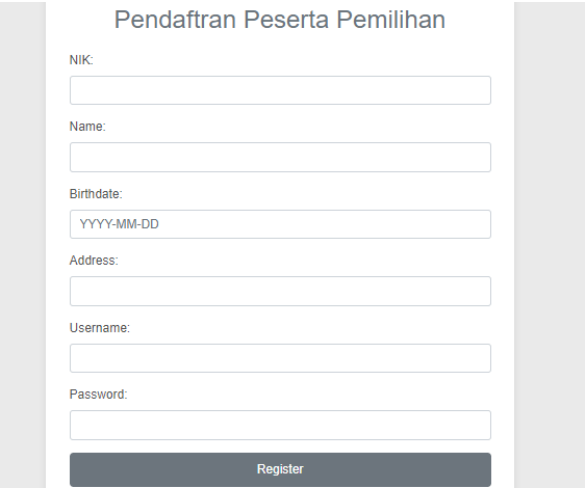
Jadi, $D(c1 \times c2) = (m1 + m2) \bmod n = 71$.

4 Hasil dan Pembahasan

Hasil dan Pembahasan meliputi Pelaksanaan Implementasi metode Kriptografi Homomorfik di Sistem Web E-Voting terbagi menjadi 2 yaitu Implementasi Anonimisasi Data dan Kriptogafi Homomorfik dan pengujian keamanan dilakukan untuk metode Kriptografi Homomorfik berdasarkan pengujian Chipher-text Only (COA) Testing dan Shannon Entropy

4.1 Anonimisasi Data

Hasil dari proses anonimisasi data yang digunakan oleh petugas untuk memasukkan data diri peserta pemilih, seperti NIK, nama, tanggal lahir, alamat, username, dan password. Username dan password dihasilkan secara manual oleh petugas. Setelah data diinput, data tersebut akan dianonimkan dan dienkripsi sebelum disimpan di dalam database. Halaman ini digunakan oleh petugas untuk memasukkan data diri peserta pemilih, seperti NIK, nama, tanggal lahir, alamat, username, dan password. Username dan password dihasilkan secara manual oleh petugas. Setelah data diinput, data tersebut akan dianonimkan dan dienkripsi sebelum disimpan di dalam database. Tampilan halaman registrasi dapat dilihat pada Gambar 11 berikut :



Gambar 11 Proses Anonimisasi Input Data

Tabel 1. Menunjukan hasil tersimpan ke database secara terenkripsi nik,username,password dari proses anonimisasi menginputkan data diri

Tabel 1 Hasil Proses Enkripsi Homomorfik

nik	username	pwd
aB12x9Yz03PQrLv58FJkCd30Xy	aG9iN0xq83dQWyz	Rt45PzX1jKwH7Co
R8xZ5tNq4MkLrC0yP9aJZwv12	nTv92Lp3MqW8sXo	Yk67DfR2pNzHwXa
pQrZ83JkFy0xNCLv91MwYzT58	pX34KmNv81RyTzL	Qr56TyNzX3WvLpA
xL9MwYJ58pRkC0TqZrN3vaF21	zXo1jKwRtP83LvM	LpX1KmNvTy27WqH
LJv91MZr8N5kCxMwT3pFaY09q	NzXoLp34KmP8TyW	WvL3NzTyKm57PqR
YFJrN3vLpZ5kCq9Mw08aT91xM	PqW3XNzvLp71TyK	TyX34oNzKm8PqWL
MwYzTqR3p9Lv58N0JkCFaZ8xM	NzXoTyKmPqL1W34	WvL3XoNz57TyKmP
p8NqR5kCJY91ZLv03rMwTzaF8M	KmTyNz34oLpW8Pq	LpWvNzXKmTy57qRo
F8Jv9N3pMwR5kZCLq8aT1Yz0L	NzXKmPqW3oTyL34	WvL3NzKm57TyPqXR
zT91LvMwR5kCqYF3p8aN0JZ8rM	TyKmPqNz34oW8LX	WvNzKmTy57PqRoX1
MwTq8aJZLv91R5pN3kYFC0L8zr	KmNzXoTyPqL1W34	PqL3XoTyNzKm57WR
q8LvJZR5pMwT3kYFC01zaN9Mx8	WvTyKmPqNz34oLX	NzXoLpKmTyPqWv57
R5LvZkYFC0q8MwT3Jp9aN01zr8	XoPqNzKmTyL1W34	WvL3NzKmTy57PqRo
T3pMwNq8LvZkJYFC0r5aN9Mx8z	NzXKmTyPqL34oW8	TyNzKmXoPqWv57L3
LvZkJ9YFC0Nq8MwT3pR5aN8xzM	WvL3XoNzKmTyPq71	PqLNzTyKmW8Xo34vR

4.2 Enkripsi

Proses enkripsi dilakukan pada kandidat vote nomor pasangan calon kandidat yaitu “1” dan “2” Gambar 12 menunjukan bahawa terdapat plaintext “1” dan “2”

Gambar 12 Plaintext

Tabel 2 Hasil Proses Enkripsi Homomorfik

encrypted_vote	**kandidat**
204958203948573204958203948573201 9	2147483647
583204958203948573201948573204958 2	2147483647
482039475830294857320948573201948 5	2147483647
582039485730294857320948573204958 3	2147483647
730294857320948573204582930294857 3	2147483647
812304596041203894582730459203487 6	2147483647
459302184730458203947583201948573 0	2147483647
320948573204584730291847583049281 0	2147483647
758203948573201948320948573029847 5	2147483647
573201948573021938475820394758203 9	2147483647
948573204582730948320184573204587 3	2147483647
394758203948573201948573204582730 9	2147483647
204958203947582030291847583204583 0	2147483647
203948573029184730459320184759203 8	2147483647
582304573201948573204958203947582 0	2147483647

Gambar 12 menunjukan hasil dari Proses enkripsi dilakukan pada kandidat vote nomor pasangan calon kandidat yaitu "1" dan "2", dari Gambar dibawah ini terdapat 2 peserta yang telah memilih calon kandidatnya. Sedangkan tTabel 2 Memperlihatkan hasil dari proses enkripsi. Setelah peserta melakukan pemilihan, suara yang diberikan untuk kandidat akan tersimpan dalam database dalam bentuk terenkripsi. Proses anonimisasi menunjukkan bahwa data peserta berhasil dianonimkan setelah suara mereka tercatat.

4.3 Proses Pengolahan Data Homomorfik

Hasil pengolahan data dengan metode Homomorfik menunjukkan bahwa meskipun data suara yang masuk telah terenkripsi menjadi ciphertext, proses penjumlahan suara tetap dapat dilakukan. Hasil akhirnya menunjukkan total 2 suara, dengan masing-masing kandidat menerima 1 suara.

Gambar 13 Proses Homomorfik

4.4 Dekripsi

Hasil dekripsi yang dilakukan setelah melalui tahapan pembentukan kunci, enkripsi, pengolahan homomorfik, dan akhirnya dekripsi menunjukkan bahwa pengolahan data dengan metode Homomorfik tetap dapat menjaga kerahasiaan data suara. Meskipun data suara terenkripsi selama seluruh proses pengolahan, sistem mampu menghasilkan hasil akhir yang akurat. Data suara yang masuk, meskipun dalam bentuk terenkripsi, berhasil diolah tanpa perlu didekripsi terlebih dahulu, dan saat proses dekripsi terakhir dilakukan, hasilnya sesuai dengan suara yang diberikan oleh peserta, menunjukkan bahwa sistem e-voting ini mampu menjaga privasi dan akurasi penghitungan suara.

Tabel 3 Hasil Proses Dekripsi Homomorfik

Id vote	kandidat
1	2
2	2
3	2
4	1
5	2
6	1
7	2
8	1
9	1
10	1
11	2
12	1
13	2
14	2
15	1

4.5 Pengujian Cipher-Text Only (COA)

Selain proses dekripsi, dilakukan pula pengujian menggunakan metode cipher-text only attack (COA) untuk menilai kekuatan sistem enkripsi berbasis algoritma Paillier. Pada tahap ini, fungsi coa_test diimplementasikan untuk menguji potensi serangan terhadap ciphertext. Pengujian ini bekerja dengan menerima input ciphertext dari pengguna melalui metode POST dan mencoba menebak plaintext yang sesuai melalui teknik brute-force. Jika dalam batas tebakan tertentu ciphertext berhasil dipecahkan, plaintext akan ditampilkan. Namun, jika serangan gagal menemukan kecocokan dalam jumlah tebakan maksimum yang ditentukan, sistem akan menginformasikan bahwa ciphertext tidak dapat didekripsi dalam batasan tersebut.

Tabel 4 Hasil Pengujian Cipher Text Only (COA)

Ciphertext	Algoritma	Hasil	Tingkat Keberhasilan
191220405421905949302020304959403O	Pailler	Tidak Ditemukan	0%
8123045960412038945827304592034876	Pailler	Ditemukan Plaintext	0%
459302184730458203947583201948573O	RSA	52	30%
320948573204584730291847583049281O	Pailler	Tidak Ditemukan	0%
758203948573201948320948573029847S	RSA	12	10%
203948573029184730459320184759203S	RSA	70	56%
573201948573021938475820394758203O	Pailler	Tidak Ditemukan	0%
948573204582730948320184573204587S	RSA	5	
204958203947582030291847583204583O	Pailler	Tidak Ditemukan	0%
582304573201948573204958203947582O	Pailler	Tidak Ditemukan	0%
394758203948573201948573204582730S	RSA	30	45%
482039475830294857320948573201948S	Pailler	Tidak Ditemukan	0%
582039485730294857320948573204958S	RSA	80	70%
730294857320948573204582930294857S	RSA	8	15%

Selama serangan brute-force, setiap nilai dari 1 hingga max_guesses - 1 akan dicoba satu per satu. Setiap tebakan diuji dengan mengenkripsi nilai menggunakan kunci publik, dan hasil enkripsi dibandingkan dengan ciphertext yang diberikan. Jika ditemukan kesesuaian antara ciphertext dan hasil enkripsi, nilai yang ditebak akan ditetapkan sebagai decrypted_value, dan variabel found akan diatur menjadi True.

Hasil dari pengujian brute-force ini ditampilkan pada Tabel 4, yang menunjukkan apakah sistem mampu mempertahankan integritas enkripsinya atau berhasil dipecahkan dalam serangan tersebut. Hasilnya memberikan wawasan mendalam mengenai kekuatan keamanan sistem terhadap serangan cipher-text only, sekaligus mengukur seberapa efektif algoritma Paillier dalam menjaga kerahasiaan data terenkripsi. aillier: Pengujian menunjukkan bahwa algoritma Paillier tetap aman terhadap metode ciphertext-only attack pada semua sampel yang diuji. Ini menegaskan tingginya tingkat keamanan algoritma ini terhadap serangan berbasis ciphertext.

Algoritma RSA memiliki tingkat kerentanan yang lebih tinggi dibandingkan Paillier, dengan keberhasilan serangan mencapai 10-70% pada pengujian dengan 14 sampel chiphertext. Kerentanan ini disebabkan oleh kelemahan RSA yang berkaitan dengan analisis faktorisasi bilangan besar.

4.6 Pengujian Shannon Entropy

Selama pengujian Shannon entropy, yang digunakan untuk menilai kekuatan hasil enkripsi, dilakukan pengukuran terhadap seberapa acak dan kuat ciphertext tersebut. Pengujian ini memanfaatkan alat yang tersedia dari situs Shannon entropy calculator. Prosesnya melibatkan input

ciphertext yang telah terenkripsi ke dalam kalkulator, kemudian nilai entropy yang dihasilkan diinterpretasikan untuk mengevaluasi ketahanan ciphertext. Sebagai contoh, seperti yang terlihat pada Gambar, hasil enkripsi memiliki nilai entropy sebesar 3.20 bits. Nilai entropy sebesar 3.20 bits menunjukkan bahwa rata-rata setiap karakter dalam ciphertext membawa 3.20 bits informasi. Ini mencerminkan tingkat variasi dan ketidakpastian dalam data terenkripsi. Semakin tinggi nilai entropy, semakin acak dan bervariasi data tersebut, sehingga hasil enkripsi dianggap cukup kuat dan efektif dalam menjaga keamanan data.

Tabel 5. Hasil Pengujian Shannon Entropy

Ciphertext	Algoritma	Entropy	Tingkat Keacakan
191220405421905949302020304959403O	Pailler	8.7	0%
8123045960412038945827304592034876	Pailler	5.8	0%
459302184730458203947583201948573O	RSA	2.1	30%
320948573204584730291847583049281O	Pailler	6.5	0%
758203948573201948320948573029847S	RSA	3.2	10%
203948573029184730459320184759203S	RSA	2.5	56%
573201948573021938475820394758203S	Pailler	5.1	0%
948573204582730948320184573204587S	RSA	4.1	
204958203947582030291847583204583O	Pailler	5.3	0%
582304573201948573204958203947582O	Pailler	6.2	0%
394758203948573201948573204582730S	RSA	4.2	45%
482039475830294857320948573201948S	Pailler	6.6	0%
582039485730294857320948573204958S	RSA	3.8	70%
730294857320948573204582930294857S	RSA	2.5	56%

Nilai rata-rata entropi yang sangat tinggi (mendekati angka maksimum 8) menunjukkan bahwa tingkat keacakan algoritma ini sangat baik. Hal ini membuktikan bahwa ciphertext yang dihasilkan oleh Paillier sangat sulit ditebak atau dipecahkan. Entropi yang dihasilkan RSA lebih rendah dibandingkan Paillier dan AES, menunjukkan adanya pola tertentu dalam ciphertext RSA yang dapat dimanfaatkan dalam serangan.

Keunggulan Paillier berdasarkan pengujian, algoritma Paillier terbukti memiliki tingkat keamanan lebih tinggi daripada RSA dan memiliki tingkat entropi yang sebanding dengan AES. Keunggulan ini menjadikan Paillier sebagai pilihan yang sangat cocok untuk aplikasi seperti e-voting, di mana anonimitas dan keamanan data pemilih adalah hal yang sangat penting. RSA memiliki kelemahan yang signifikan terhadap serangan ciphertext-only, yang dapat menjadi ancaman serius jika tidak dilengkapi dengan langkah mitigasi tambahan. Hal ini menggarisbawahi bahwa RSA kurang sesuai untuk aplikasi yang memerlukan tingkat keamanan tinggi.

Penelitian ini menunjukkan bahwa algoritma homomorfik seperti Paillier tidak hanya unggul dalam mendukung perhitungan langsung pada data terenkripsi, tetapi juga memberikan perlindungan yang lebih baik terhadap serangan ciphertext-only. Oleh karena itu, penelitian ini memberikan kontribusi yang penting dalam pengembangan sistem e-voting yang lebih aman dan efisien.

5 Kesimpulan

Kesimpulan dari penelitian ini menunjukkan bahwa penggunaan Kriptografi Homomorfik Paillier untuk anonimisasi data terenkripsi dalam sistem e-voting berbasis Flask Python terbukti cukup aman. Hal ini didasarkan pada dua jenis pengujian:

1. Kesesuaian Hasil: Pengujian ini memastikan bahwa hasil yang diperoleh sesuai dengan yang diharapkan.
2. Pengujian Ciphertext Only: Tidak ditemukan plaintext yang cocok dan ciphertext tidak dapat dibaca.

Ucapan Terimakasih

Selama penelitian ini berlangsung, penulis menghadapi berbagai tantangan dan kendala. Namun, berkat bantuan dan berkat dari Allah SWT, serta dukungan dari berbagai pihak, penulis dapat menyelesaikan penelitian ini. Dengan rasa syukur yang mendalam, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua kalangan masyarakat yang ingin membaca hasil penelitian dan dapat di kembangkan untuk kedepannya

Daftar Pustaka

[1] K. A. A. P. N. A. Cahya Rahmad, "Noble Method for Data Hiding using Steganography Discrete Wavelet Transformation and Cryptography Triple Data Encryption Standard: DES".

[2] L. M. P. Fausto Neri da Silva Vanin, "A Blockchain-Based End-to-End Data Protection Model for Personal Health Records Sharing: A Fully Homomorphic Encryption Approach," *Sensor,* 2023.

[3] J. R. Khairul Muttaqin, "Analysis And Design Of FIle Security System AES (Advanced Encryption Standard) Cryptography Based," *Journal of Applied Engineering and Technological Science,* 2020.

[4] M. D. E.-C. E. K. Ahmed EL-YAHYAOUI, "A verifiable fully homomorphic encryption scheme to secure big data in cloud computing," *IEEE,* 2017.

[5] F. A. S. M. Nur Muhammad Dwi Oktafiansyah, "Penerapan Kriptografi Dengan Algoritma Data Encyption Standart Pada Text Hasil Konversi Dari Citra," *e-journal Unmul,* vol. 1, 2016.

[6] C. S.-a. S. K. Kulkatechol Kanokngamwitroj, "The effect of data anonymization on a data science project," *IEEE,* 2022.

[7] J. B. F. B. Ahmad Al Badawi, "Open-Source Fully Homomorphic Encryption Library," *WAHC'22: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography,* 2022.

[8] R. Munir, "IF4020 Kriptografi Enkripsi Homomorfik," 2020. [Online]. [Diakses 13 3 2020].

[9] R. D. A. Aditya Maulana Rajak, "PURWARUPA SISTEM E-VOTING MENGGUNAKAN ENKRIPSI HOMOMORPHIC DI KOMISI PEMILIHAN UMUM KOTA BANDUNG," *Jurnal Penelitian Mahasiswa Teknik Dan Ilmu Komputer,* vol. 1, 2021.

[10] E. B. P. B. B. B. Irma T. Plata, "A Security Approach for File Management System using Data Encryption Standard (DES) algorithm," *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 08, 2019.

[11] D. I. R. M. S. M. Muhtar Hartopo, "Pengembangan Aplikasi E-Voting Menggunakan Enkripsi Homomorfik," *Makalah TA Muhtar Hartopo,* 2022.

[12] D. B. T. Saja J Mohammed, "Privacy Preserving Algorithm using Chao Scattering of Partial Homomorphic Encryption," *Journal of Physics: Conference Series,* 2021.

[13] "Mengenal Apa Itu Enkripsi serta Jenis, Cara Kerja, dan Manfaatnya," *CLOUDMATIKA,* vol. 25, p. 10, 2022.

[14] Tony, "Implementasi Paillier Cryptosystem," *Makalah ke-2 IF4020 Kriptografi, Semester II.*

[15] R. Munir, "IF4020 Kriptografi Enkripsi Homomorfik," 2020. [Online]. [Diakses 13 3 2024].

[16] D. B. T. Saja J Mohammed, "Privacy Preserving Algorithm using Chao Scattering of Partial Homomorphic Encryption," *Journal of Physics: Conference Series,* 2021.