JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 12 No. 3S1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v12i3S1.5065

IMPLEMENTASI ALGORITMA RSA PADA APLIKASI E-VOTING (STUDI KASUS: ORAGANISASI MAHASISWA UMKT)

Dery Dinata[1], Sayekti Harits Suryawan[2*], Muhammad Taufiq Sumadi[3]

[1],[2]Teknik Informatika, Fakultas Sains dan teknologi, Universitas Muhammadiyah Kalimantan Timur, Kota Samarinda, Kalimantan Timur,
[3] Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Samarinda, Kota Samarinda, Kalimantan Timur.

Received: 26 Juli 2024
Accepted: 5 Oktober 2024
Published: 12 Oktober 2024

Keywords:
E-Voting
Keamanan Data
Kriptografi
Algoritma RSA

Corespondent Email:
shs500@umkt.ac.id

Abstrak. E-voting adalah metode pemungutan suara digital yang mencakup seluruh proses dari pendaftaran hingga pengiriman hasil. E-voting telah menjadi topik yang populer di seluruh dunia, termasuk di Indonesia. Walaupun memiliki banyak keuntungan seperti efisiensi waktu dan biaya serta peningkatan akurasi, e-voting rentan terhadap serangan *cyber* yang dapat membahayakan integritas dan kerahasiaan data. Penelitian ini bertujuan mengamankan data suara dalam database e-voting menggunakan metode kriptografi dengan algoritma RSA. Hasil pengujian menunjukkan bahwa penerapan algoritma RSA berhasil meningkatkan keamanan data suara, sehingga data terenkripsi tidak dapat diakses atau diubah oleh pihak yang tidak berwenang. Dengan demikian, algoritma RSA terbukti dapat diandalkan untuk memberikan perlindungan yang kuat terhadap data suara dalam sistem aplikasi e-voting.

Abstract. *E-voting is a digital voting method that encompasses the entire process from registration to result transmission. E-voting has become a popular topic worldwide, including in Indonesia. Despite its many advantages, such as time and cost efficiency and improved accuracy, e-voting is vulnerable to cyber attacks that can threaten data integrity and confidentiality. This study aims to secure voice data in e-voting databases using cryptographic methods with the RSA algorithm. Test results show that the implementation of the RSA algorithm successfully enhances the security of voice data, making encrypted data inaccessible or unalterable by unauthorized parties. Thus, the RSA algorithm proves to be reliable for providing strong protection for voice data within e-voting applications.*

1. PENDAHULUAN

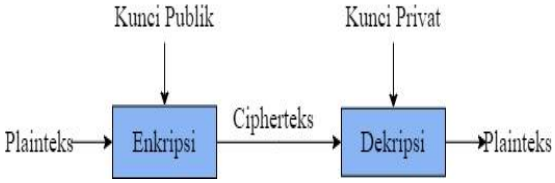
E-voting adalah pelaksanaan pemungutan suara secara digital, yang meliputi proses dari pendaftaran, pelaksanaan, penghitungan, hingga pengiriman hasil suara. E-voting telah menjadi topik yang populer di seluruh dunia, termasuk di Indonesia. Di era digital saat ini, beberapa negara telah mengadopsi e-voting sebagai metode untuk pemilihan presiden atau pemimpin organisasi [1]. E-voting yang dikenal luas oleh masyarakat biasanya digunakan dalam Pemilu (Pemilihan Umum) atau Pilkada (Pemilihan Daerah). Namun, e-voting skala kecil juga digunakan untuk pemilihan presiden atau ketua organisasi internal kampus dan perguruan tinggi lainnya [2]. E-voting memiliki beberapa keunggulan, seperti mempercepat proses pemilihan, mengurangi biaya, dan meningkatkan akurasi dalam pemilihan suara. E-voting juga memudahkan pemilih untuk

menggunakan hak pilihnya tanpa harus menunggu antrian yang lama [3-4].

Namun, sistem e-voting menghadapi tantangan terutama dalam hal integritas keamanan dan kerahasiaan data. Karena e-voting beroperasi secara online, sistem ini rentan terhadap serangan *cyber* yang dapat memanipulasi dan membocorkan data. Oleh karena itu, penerapan keamanan pada aplikasi e-voting sangat penting untuk melindungi dari serangan *cyber* [1]. Keamanan adalah aspek penting dalam melindungi sistem, karena tanpa keamanan, sistem akan menjadi target empuk bagi peretas. Untuk memastikan data dalam sistem tetap aman, diperlukan metode yang efektif untuk mengatasi masalah ini [5]. Salah satu metode yang dapat digunakan adalah metode kriptografi.

Kriptografi adalah metode teknologi informasi yang digunakan untuk mengamankan data pribadi atau rahasia [6]. Kriptografi terdiri dari dua jenis algoritma, simetris dan asimetris. Contoh algoritma simetris *Data Encryption Standard* (DES) dan *Advanced Encryption Standard* (AES), sedangkan contoh algoritma asimetris termasuk *Rivest Shamir Adleman* (RSA) dan *Digital Signature Algorithm* (DSA) [7]. Algoritma simetris menggunakan satu kunci yang sama untuk enkripsi dan dekripsi, sementara algoritma asimetris menggunakan dua kunci berbeda untuk enkripsi dan dekripsi [8].

Dalam penelitian ini menggunakan metode kriptografi dengan algoritma RSA. Algoritma RSA adalah salah satu teknik kriptografi dimana kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi. Algoritma RSA dikembangkan pada tahun 1977 oleh tiga peneliti dari MIT *(Massachusetts Institute of Technology)*, yaitu Ron Rivest, Adi Shamir, dan Leonard Adleman. Nama RSA merupakan inisial dari nama belakang ketiga peneliti tersebut [9]. Kunci yang digunakan untuk enkripsi adalah kunci publik, sedangkan kunci yang digunakan untuk dekripsi adalah kunci privat [10]. Algoritma RSA memiliki keunggulan dalam hal tingkat keamanan yang tinggi karena menggunakan dua kunci yang berbeda untuk proses enkripsi dan dekripsi [11]. Berikut adalah gambar proses enkripsi dan dekripsi algoritma asimetris pada gambar 1.



Gambar 1 Proses Algoritma Asimetris

Algoritma RSA mempunyai tahap proses perhitungan yang meliputi tiga tahapan, pembangkitan kunci *(generate key)*, enkripsi *(encryption)*, dan dekripsi *(decryption)* [12].

1.1. Tahapan Pembangkitan kunci

1. Pilih dua bilangan prima yang besar p dan q. Nilai p dan q bersifat rahasia (privat)
2. Hitung nilai $n = p \times q$. Nilai n tidak dirahasiakan sebaiknya $p \neq q$. Karena jika $p = q$ maka $n = p2$ sehingga p didapatkan dengan akar pangkat dua dari n
3. Menghitung $\varphi(n) = (p-1)(q-1)$
4. Memilih kunci publik yang disebut e, relatif prima terhadap φ, artinya faktor pembagi keduanya adalah 1, yang disebut secara matematika $gcd\ (e, \varphi) = 1$
5. Menghitung kunci privat (dekripsi) menggunakan rumus $e.d\ mod\ n = 1$
6. Maka hasil pembentukan kunci publik dan kunci privat adalah (e,n) untuk kunci publik dan (d,n) untuk kunci privat.
7. Nilai n tidak bersifat rahasia karena diperlukan pada saat perhitungan proses ekripsi dan deskripsi.

1.2. Tahapan Enkripsi

1. Masukan nilai hasil plainteks.
2. Konversi dalam bentuk $UTF-8$
3. Masukan kunci publik (e,n)
4. Lakukan perhitungan dengan rumus $C = M\text{^}e\ mod\ n$
5. Menemukan cipherteks.

1.3. Tahapan Dekripsi

1. Masukan pesan cipherteks yang telah ditemukan.
2. Masukan kunci privat (d,n)
3. Lakukan perhitungan dengan rumus $P = C\text{^}d\ mod\ n$
4. Konversi dalam bentuk $UTF-8$
5. Menemukan hasil deskripsi.

Algoritma RSA dianggap aman karena menggunakan konsep matematika yang melibatkan bilangan besar sebagai faktor prima. Semakin besar angka prima yang digunakan, semakin baik keamanan data pada sistem tersebut [13].

Penelitian Sebelumnya yang pernah dilakukan Munir, RSA akan tetap aman jika modulus n cukup besar. Jika panjang n hanya 256 bit atau kurang, angka tersebut dapat difaktorkan dalam beberapa jam dengan komputer/PC, dan jika panjang n adalah 512 bit atau kurang, dapat difaktorkan dengan beberapa ratus komputer. Saat ini panjang kunci RSA yang aman adalah 2048 bit [14].

Penelitian sebelumnya telah menunjukkan bahwa algoritma RSA dapat digunakan secara aman dan efektif dalam pengembangan sistem e-voting. Penelitian yang dilakukan oleh Setiawan, RSA sebagai algoritma kriptografi yang terkemuka, memberikan perlindungan yang kuat terhadap data yang dikirim dan disimpan dalam database [13].

Penelitian yang dilakukan oleh Anggoro dalam membuat sistem keamanan dengan algoritma RSA untuk menjamin kerahasian data hasil pemilihan. Hasil dari penelitian bahwa algoritma RSA dapat digunakan untuk menjaga kerahasiaan data hasil pemilihan pada aplikasi e-voting [15].

Penelitian oleh Putra juga mendukung hal ini dengan menunjukkan bahwa kombinasi antara RSA dan base64 dapat meningkatkan keamanan dan kerahasiaan data secara signifikan. Kombinasi ini memberikan lapisan tambahan yang penting, membuat sistem e-voting lebih dapat bertahan terhadap berbagai serangan *cyber* [16].

Berdasarkan penjelasan uraian diatas, penelitian ini fokus pada kerentanan data dalam database pada aplikasi e-voting serta penerapan metode kriptografi algoritma RSA untuk mengatasi masalah tersebut. Algoritma RSA dipilih untuk menjaga keamanan dan kerahasiaan data dalam database aplikasi e-voting. Dengan menerapkan algoritma RSA, data akan dienkripsi menggunakan kunci publik sebelum dikirimkan, dan hanya bisa didekripsi oleh sistem yang memiliki kunci privat untuk dekripsi berdasarkan id pemilih.

2. Tinjauan Pustaka

Berbagai peneltitan terdahulu yang berkaitan dengan topik dan metode yang sedang dibahas atau digunakan membantu memberikan pemahaman dan perspektif yang lebih dalam mengenai penelitian ini.

Pada penelitian terdahalu yang dilakukan oleh Susanto [17] penelitian ini berhasil mengembangkan aplikasi keamanan pesan teks yang efektif menggunakan algoritma RSA, memberikan solusi yang aman dan praktis untuk melindungi kerahasiaan pesan teks yang dikirim melalui SMS.

Pada penlitian yang dilakukan oleh Hasbulloh [18] penilitian ini berhasil mengembangkan dan menerapkan sistem e-voting berbasis web dengan menggunakan algoritma RSA untuk meningkatkan keamanan dan efisiensi dalam pemilihan organisasi ikatan pondok pesantren *Smart- SIPKOTREN.*

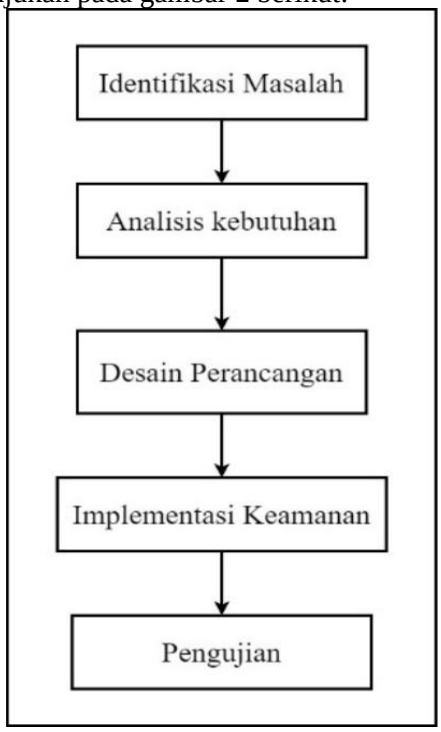
Pada penelitian yang dilakukan oleh Andriani [19] peneitian ini berhasil menerapkan algoritma RSA efektif dalam meningkatkan keamanan data penjualan dan memberikan perlindungan terhadap ancaman pencurian dan manipulasi data di Toko Baju *Familly.*

Pada penelitian yang dilakukan oleh Farhan [20] merancang sistem keamanan dengan menerapkan algoritma RSA. Sistem tersebut mampu menjaga dan meningkatkan keamanan data penjualan di database MySQL Perum Bulog Kanwil Sumatera Utara dan sistem yang dirancang dapat membantu menjaga kerahasiaan data dari pihak yang tidak berwenang.

3. METODE PENELITIAN

Penelitian ini menggunakan metode campuran *(mix method),* menggabungkan pendekatan kuantitatif eksperimental untuk membuat sistem keamanan data suara dalam database aplikasi e-voting menggunakan algoritma RSA, pendekatan kualitatif melalui studi kasus di Universitas Muhammadiyah Kalimantan Timur (UMKT) dengan melakukan wawancara pada narasumber yang terkait di UMKT untuk mendapatkan pemahaman mendalam tentang kebutuhan pengguna. Melalui eksperimen ini, peneliti menguji sistem dengan berbagai skenario. Penelitian bertujuan menerapkan algoritma RSA untuk menjaga keamanan dan kerahasiaan data suara pada sistem e-voting. Terdapat lima tahapan penelitian dimulai dari identifikasi masalah,

analisis kebutuhan, desain perancangan, implementasi kemanan dan pengujian. Seperti ditunjukan pada gambar 2 berikut:



Gambar 2 Diagram Alur

3.1. Identifikasi Masalah

Penelitian bertujuan untuk mengatasi masalah keamanan data yang sering terjadi pada sistem aplikasi e-voting. Dalam mengidentifikasi masalah peneliti melakuakan studi literatur dengan membaca, mempelajari dan mengumpulkan jurnal-jurnal serta referensi lainnya untuk mendapatkan informasi dalam mengatasi permasalahan tersebut. Hasil dari permasalahan yang sudah terjadi adalah dari segi integritas keamanan dan kerahasiaan data yang rentan terhadap manipulasi oleh pihak yang tidak bertanggung jawab.

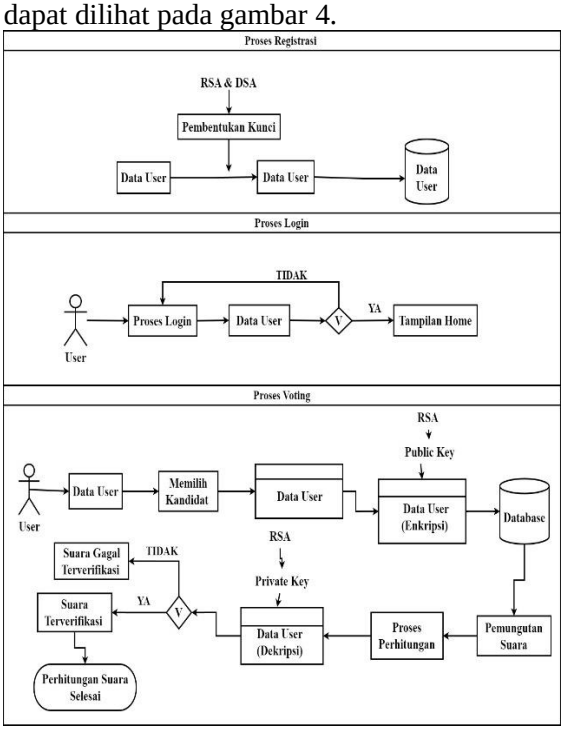
3.2. Analisis Kebutuhan

Setelah mengidentifikasi masalah, peneliti melakukan wawancara dengan mahasiswa Universitas Muhammadiyah Kalimantan Timur. Tujuan dari wawancara ini adalah untuk memahami kebutuhan yang harus dipenuhi dalam sistem e-voting tersebut, termasuk kebutuhan pengguna, fitur-fitur keamanan, dan kebutuhan lain yang relevan. Berikut adalah gambar dari hasil wawancara bersama narasumber dilihat pada gambar 3.

Gambar 3 Wawancara Bersama Narasumber

3.3. Desain Perancangan

Setelah menganalisis kebutuhan, peneliti merancang desain sistem. Tahap ini melibatkan desain alur keamanan data suara pada sistem e-voting menggunakan algoritma RSA. Desain ini mencakup dari alur kerja RSA pada sistem, pembangkitan kunci publik dan kunci privat dan mekanisme proses enkripsi dan dekripsi untuk menyimpan dan menampilkan data suara. Berikut adalah perancangan *grand* desain RSA dapat dilihat pada gambar 4.



Gambar 4 *Grand* Desain RSA

Diatas adalah grand desain RSA, peran RSA didalam sistem tersebut adalah untuk mengamankan data suara sebelum tersimpan dalam database.

3.4. Implemantasi Keamanan

Setelah merancang desain sistem, langkah selanjutnya adalah mengimplementasikan keamanan menggunakan algoritma RSA. Peneliti akan menerapkan desain yang telah dibuat untuk menjaga keamanan data suara dalam database sistem e-voting. Implementasi dilakukan menggunakan spek perangkat prosesor AMD Ryzen™ 5 5625u 6 core 12 thread, RAM 8GB, penyimpanan 256GB SSD, dengan penggunaan bahasa pemrograman *Python* dan *Framework Django* untuk menuliskan kode-kode yang dibutuhkan.

3.5. Pengujian

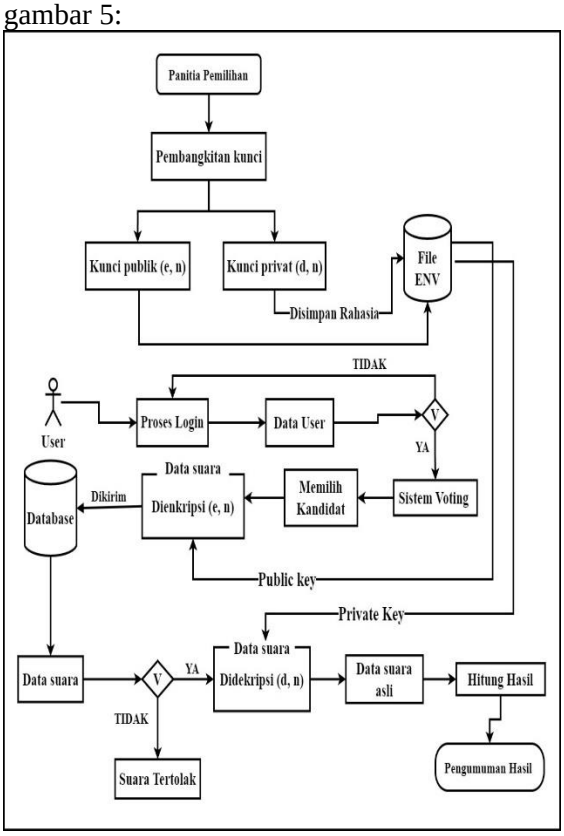
Setelah selesai melakukan implementasi, sistem akan diuji untuk memastikan keamanannya sesuai dengan harapan. Pengujian terdiri dari pengujian fungsionalitas dan pengujian sistem. Pengujian fungsionalitas bertujuan untuk memastikan bahwa proses enkripsi dan dekripsi data berjalan dengan benar, di mana data yang telah dienkripsi dapat didekripsi kembali dengan hasil yang sama persis seperti data aslinya. Pengujian sistem meliputi verifikasi suara, di mana sistem diuji dengan mencoba mengubah nilai data suara langsung di database untuk melihat apakah sistem masih dapat mengonfirmasi keaslian data tersebut.

4. HASIL DAN PEMBAHASAN

4.1. Desain Perancangan

4.1.1. Alur Kerja Algoritma RSA pada Sistem

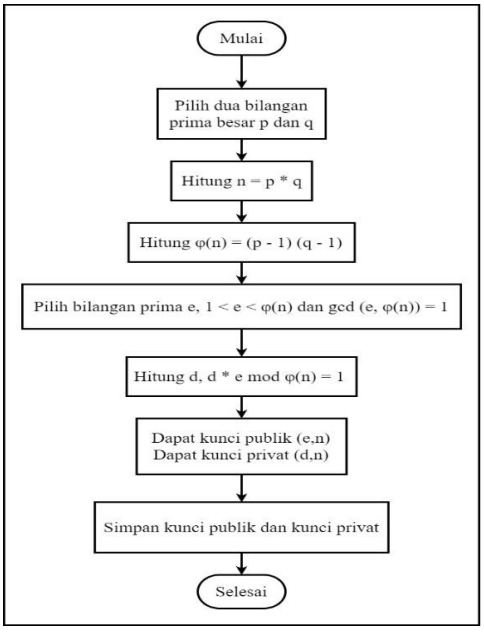
Proses dimulai dengan panitia pemilihan yang bertanggung jawab untuk menghasilkan kunci publik (e, n) dan kunci privat (d, n). Kunci publik digunakan untuk enkripsi data, sementara kunci privat digunakan untuk dekripsi data. Kunci tersebut disimpan secara rahasia dalam file ENV. *User* memulai proses dengan login ke sistem pemilihan. Setelah berhasil login, data pengguna diverifikasi. Jika verifikasi gagal, proses ditolak dan *user* kembali ke proses login. *User* yang berhasil login kemudian melanjutkan untuk memilih kandidat di sistem voting. Data suara *user* memilih kandidat, dienkripsi menggunakan kunci publik dan dikirim ke database. Data suara yang diterima diperiksa keasliannya. Jika data tidak *valid*, data suara tersebut ditolak. Data suara yang *valid* akan didekripsi menggunakan kunci privat untuk memperoleh data suara asli. Data suara asli kemudian dihitung untuk menghasilkan hasil akhir pemilihan. Setelah proses penghitungan selesai, hasil pemilihan diumumkan. Berikut adalah gambar alur kerja RSA pada sistem dilihat pada gambar 5:



Gambar 5 Alur Kerja RSA pada Sistem

4.1.2. Pembangkitan Kunci

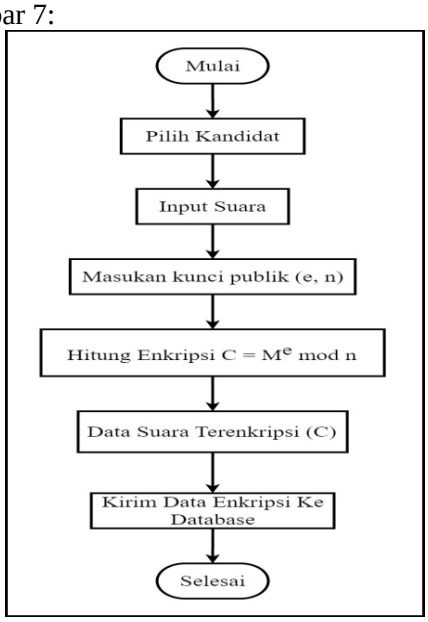
Pembangkitan kunci merupakan proses awalan dalam algoritma RSA. Proses pembangkitan kunci tahap pertama yang dilakukan dengan cara memasukan angka prima besar nilai p dan q. Dimana nilai dari p dan q menjadi nilai dari n dan $\varphi(n)$. Hitung nilai e (enkripsi), maka nilai e berfungsi sebagai kunci publik terhadap (n). Lalu hitung nilai d (dekripsi) untuk mencari kunci privat terhdap (n). Sehingga mendapatkan kunci publik (e, n) dan kunci privat (d, n). Berikut adalah *flowchart* Proses pembangkitan kunci dapat dilihat pada gambar 6:



Gambar 6 Proses Pembangkitan Kunci

4.1.3. Enkripsi

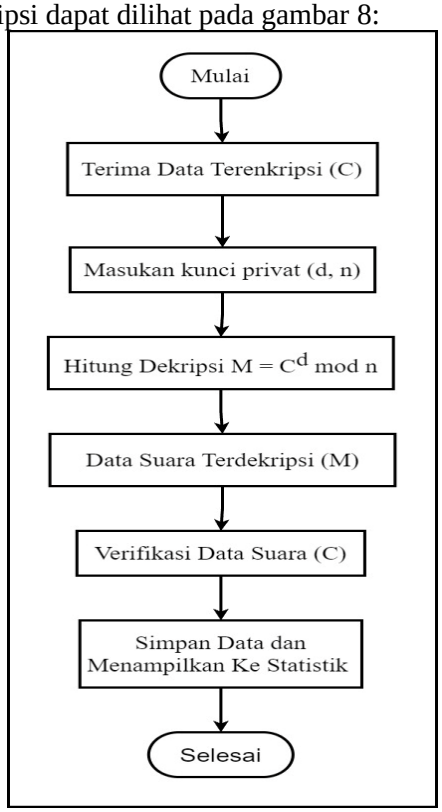
Enkripsi dilakukan setelah dilakukannya pembangkitan kunci yang akan mendapatkan kunci publik (e, n) dan kunci privat (d, n). Enkripsi dimulai dari memilih kandidat, menginput data suara, data suara berupa data yang dikonversi dalam bentuk $UTF-8$, kunci publik yang telah didapatkan diambil untuk proses perhitungan enkripsi, proses enkripsi dimana data suara asli (m) dienkripsi menjadi cipherteks (c) dan data suara yang terenkripsi disimpan kedalam database. Berikut adalah *flowchart* proses enkripsi dapat dilihat pada gambar 7:



Gambar 7 Proses Enkripsi

4.1.4. Dekripsi

Dekripsi dilakukan setelah menerima data suara yang terenkripsi (cipherteks). Sebelum melakukan dekripsi, penerima data suara harus ingat atau mengetahui kunci yang telah ditentukan antara pengirim dan penerima data. Dekripsi dimulai dari menerima cipherteks atau data suara enkripsi yang dikirim, kunci privat yang telah didapatkan diambil untuk proses perhitungan dekripsi, proses dekripsi dimana cipherteks (c) atau data suara didekripsi dalam bentuk $UTF-8$ dikembalikan menjadi data suara asli (m). Sistem melakukan verifikasi data suara yang didekripsi untuk memastikan data suara tersebut valid. Data suara yang telah diverifikasi disimpan dan ditampilkan ke statistik. Berikut adalah *flowchart* proses dekripsi dapat dilihat pada gambar 8:



Gambar 8 Proses Dekripsi

4.2. Implementasi Keamanan

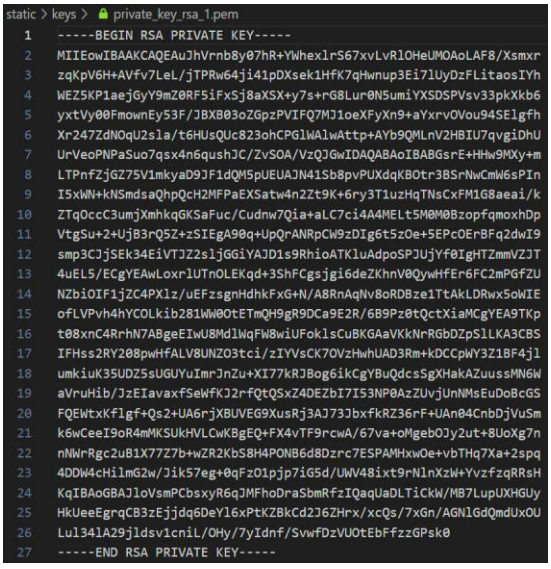
4.2.1. Pembangkitan Kunci

Hal pertama dalam menggunakan algoritma RSA adalah pembangkitan sepasang kunci publik dan kunci privat. Menghasilkan kunci tersebut peneliti membuat file bernama utilsRSA.py dalam *python.* Isi dari file tersebut terdapat code-code *python* untuk menghasilkan kunci public (e, n) dan kunci privat (d, n) yang

disimpan didalam folder *keys*. Seperti hasil *output* pada gambar 9 dan 10 berikut:

Gambar 9 *Output* Kunci Publik

Pada gambar 9 kunci publik merupakan kunci yang digunakan untuk mengenkripsi data. Kunci ini terdiri dari dua komponen publik (e) dan modulus (n). Kunci ini dapat dibagikan secara luas karena tidak dapat digunakan untuk mendekripsi data, hanya untuk enkripsi.



Gambar 10 *Output* Kunci Privat

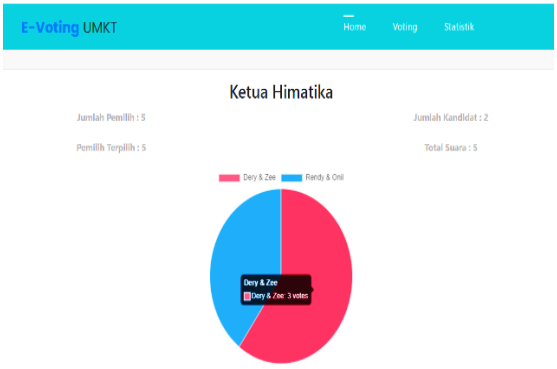
Pada gambar 10 kunci privat merupakan kunci yang digunakan untuk mendekripsi pesan yang telah terenkripsi. Kunci ini terdiri dari komponen privat (d) dan modulus (n). Kunci ini harus dijaga kerahasiaannya karena siapa pun yang memiliki kunci ini dapat mendekripsi data yang dienkripsi.

4.2.2. Mekanisme Enkripsi dan Dekripsi

Dalam mekanisme enkripsi dan dekripsi, mekanisme tersebut berada didalam satu file yang sama dengan file pembangkitan kunci sebelumnya (utilsRSA.py) hanya terdapat pada baris codingannya yang berbeda. Setiap mekanisme tersebut mempunyai codingan dan fungsi yang berbeda, sehingga menghasilkan outputnya masing-masing. Seperti pada gambar 11, 12 dan 13 berikut:

Gambar 11 *Output* Enkripsi di Database

Pada gambar 11 berikut merupakan data *voting* hasil dari enkripsi dengan kunci publik (e, n) disimpan dalam database yang hanya mengenkripsi "judul_pemilihan", "nama_kandidat" dan "nama_pemilih".



Gambar 12 *Output* Dekripsi Tampilan *Home*

Pada gambar 12 berikut merupakan tampilan statistik bagian *front* pada aplikasi e-voting, yaitu hasil dari dekripsi data suara pemilih yang telah dienkripsi dan data otomatis akan manampilkan distatistik *home.*

Gambar 13 Output Dekripsi Tampilan Dashboard

Pada gambar 13 berikut merupakan tampilan statistik bagian *back* pada aplikasi e-voting, yaitu hasil dari dekripsi data suara pemilih yang telah dienkripsi dan data otomatis akan manampilkan distatistik *dashboard.*

4.3. Pengujian

Table 1 Pengujian Fungsionalitas

NO	Deskripsi Pengujian	Langkah-Langkah Pengujian	Hasil Ynag Diharapkan	Keterangan
1	Enkripsi Data	Input data yang akan dienkripsi dengan kunci publik. Simpan data yang terenkripsi.	Data berhasil terenkripsi dengan benar.	Berhasil
2	Dekripsi Data	Ambil data yang terenkripsi yang disimpan sebelumnya. Lakukan proses dekripsi dengan kunci privat yang sesuai. Verifikasi hasil dekripsi dengan data yang asli.	Data yang dekripsi berhasil kembali sesuai dengan data asli yang telah terenkripsi.	Berhasil
3	Kesesuaian Pesan Data	Enkripsi data dengan berbagai pesan (huruf, angka, symbol, dll) menggunakan kunci publik. Dekripsi data yang terenkripsi. Memeriksa hasil dekripsi sesuai dengan format data asli atau tidak.	Data yang didekripsi sesuai dengan data asli yang terenkripsi.	Berhasil

Pengujian fungsionalitas yang telah dilakukan, sistem berhasil mengenkripsi dan mendekripsi data atau pesan dengan benar. Pengujian dilakukan dengan menghasilkan kunci publik, kemudian menggunakan kunci tersebut untuk mengenkripsi data dan mendekripsi data yang terenkripsi dengan kunci privat, data yang terenkripsi berhasil dikembalikan ke bentuk plainteks aslinya.

Table 2 Pengujian Sistem

NO	Deskripsi Pengujian	Langkah – langkah pengujian	Hasil yang diharapkan	Keterangan
1	Mengguna kan kunci privat yang sesuai	Pilih data yang sudah dienkripsi menggunakan kunci publik. Gunakan kunci privat yang sesuai untuk mendekripsi data yang telah dienkripsi.	Data asli berhasil didekripsi dengan kunci privat yang sesuai.	Berhasil
2	Mengguna kan kunci privat yang tidak sesuai	Pilih data yang sudah dienkripsi menggunakan kunci publik. Gunakan kunci privat yang tidak sesuai untuk mendekripsi data yang telah dienkripsi.	Data gagal didekripsi dengan kunci privat yang tidak sesuai dengan data asli.	Berhasil
3	Jika cipherteks dihapus 5 karakter	Pilih data yang telah dienkripsi dengan kunci piblik. Hapus 5 karakter dari cipherteks yang telah dienkripsi. Gunakan kunci privat yang sesuai untuk mendekripsi data yang telah diubah.	Data gagal didekripsi dengan kunci privat yang sesuai, hasil dekripsi berupa data yang rusak dan tidak dapat dimengerti.	Berhasil

4	Jika cipherteks tidak dihapus 5 karakter	Pilih data yang telah dienkripsi dengan kunci publik. Gunakan kunci privat yang sesuai untuk mendekripsi cipherteks tanpa menghapus 5 karakter pada cipherteks yang telah dienkripsi.	Data asli berhasil didekripsi dengan kunci privat yang sesuai tanpa menghapus 5 karakter cipherteks.	Berhasil

Pengujian sistem yang telah dilakukan sistem enkripsi dan dekripsi yang diuji berjalan dengan baik. Kunci privat yang sesuai berhasil mendekripsi data yang telah dienkripsi dengan kunci publik yang benar, kunci privat yang tidak sesuai gagal melakukan dekripsi. Penghapusan 5 karakter dari cipherteks mengakibatkan data tidak dapat didekripsi dengan benar, bahwa perubahan kecil pada cipherteks dapat merusak intergritas data. 5 karakter cipherteks yang tidak dihapus dapat didekripsi dengan benar, menunjukan keanadalan proses enkripsi dan dekripsi.

5. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma RSA berhasil meningkatkan keamanan data suara dalam database e-voting. Algoritma RSA, dengan kunci publik dan kunci privat, terbukti efektif dalam mengamankan data dari ancaman akses yang tidak sah dan manipulasi. Hasil pengujian menunjukkan bahwa data suara yang terenkripsi tidak dapat diakses atau diubah oleh pihak yang tidak berhak mengubah, sehingga menjamin integritas dan kerahasiaan data suara dalam sistem e-voting. Dengan demikian, penggunaan algoritma RSA dalam sistem e-voting ini dapat diandalkan untuk memberikan perlindungan yang kuat terhadap data suara pada sistem aplikasi e-voting.

UCAPAN TERIMA KASIH

Sebagai penulis, mengucapkan terima kasih kepada penelitian sebelumnya yang telah memberikan pengetahuan serta wawasan dalam mengenalkan metode kriptografi algoritma RSA. Kontribusi mereka memberikan inspirasi serta pemahaman yang dalam bagi penelitian ini.

Terima kasih juga kepada penemu atau pencipta algoritma RSA, inovasi ini telah memberikan kontribusi terhadap penelitian ini, sehingga penulis bisa menyelesaikan penelitian ini dengan menggunakan algoritma RSA sebagai perlindungan data.

DAFTAR PUSTAKA

[1] F. Diny Hermawati and M. Tahir, "Keamanan E-Voting Di Indonesia Melalui Pemanfaatan Kriptografi Pada Sistem AES (Advance Encryption Standard)," *Jaya Abadi Amroin*, vol. 2, no. 2, pp. 45–56, 2023.

[2] H. Angriani and Y. Saharaeni, "Implementasi Algoritma Caesar Cipher pada Keamanan Data Sistem e-voting Pemilihan Ketua Organisasi Kemahasiswaan," *Inspir. J. Teknol. Inf. dan Komun.*, vol. 9, no. 2, p. 123, 2019, doi: 10.35585/inspir.v9i2.2499.

[3] A. Yafi, P. P. Arhandi, V. A. H. Firdaus, A. Ismail, and ..., "Sistem Keamanan E-Voting Menggunakan Arsitektur Publik Blockchain Ethereum," *KLIK Kaji. Ilm. …*, vol. 4, no. 3, pp. 1313–1322, 2023, doi: 10.30865/klik.v4i3.1423.

[4] M. B. Pramadipta, " ," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4173.

[5] E. W. Rumaf, "Implementasi Algoritma Blowfish Untuk Privacy Data E-Voting," *J. Sist. Inf. Komput.*, vol. 1, no. 1, pp. 1–7, 2019.

[6] U. Ungkawa, D. Rosmala, and H. Fauzi, "Penerapan Advance Encryption Standart dalam Pengamanan Elektronik Voting," *J. Inf. Technol.*, vol. 3, no. 1, pp. 17–23, 2021, doi: 10.47292/joint.v3i1.51.

[7] Fatonah and Dadang Iskandar Mulyana, "Implementasi Metode Rivest Shamir Adleman untuk Enkripsi dan Dekripsi Text," *J. Inform. dan Teknol. Komput. (J-ICOM)*, vol. 3, no. 1, pp. 32–39, 2022, doi: 10.33059/j-icom.v3i1.4990.

[8] L. Liana, M. Zarlis, and T. Tulus, "Hybrid Cryptosystem Analysis RSA Algorithm And Triple DES Algorithm," *Sinkron*, vol. 8, no. 3, pp. 1461–1473, 2023, doi: 10.33395/sinkron.v8i3.12467.

[9] M. Rizki and P. Farida Ariyani, "Penerapan Kriptografi Dengan Menggunakan Algoritma Rsa Untuk Pengamanan Data Berbasis Desktop Pada Pt Trias Mitra Jaya Manunggal," *Skanika*,

vol. 4, no. 2, pp. 1–6, 2021, doi: 10.36080/skanika.v4i2.1991.

[10] A. Rahayu, A. P. Ardana, C. Pramudhita, D. Syafitri, and R. Z. Sirega, "Perbandingan Algoritma RSA dengan Algoritma Blowfish Pada Perancangan Aplikasi Keamanan Data," vol. 7, pp. 203–207, 2024.

[11] S. Kasus, P. Presiden, and M. Stmik, "Implementasi Kriptografi Dalam Pengamanan Database E-Voting Menggunakan Algoritma Rsa Dan Base64 Berbasis Progresive Web Apps," *e-Jurnal JUSITI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 10, no. 1, pp. 30–40, 2021, doi: 10.36774/jusiti.v10i1.818.

[12] M. S. Dairi, M. Setiani Asih, and corespondent author, "Implementasi Algoritma Kriptografi RSA Dalam Aplikasi Sistem Informasi Perpustakaaan Implementation Of RSA Cryptographic Algorithms in Library Information System Applications," *Januari*, vol. 2023, no. 2, pp. 214–223, 2022, [Online]. Available: https://jurnal.unity-academy.sch.id/index.php/jirsi/index%0Ahttp://creativecommons.org/licenses/by-sa/4.0/

[13] D. Setiawan, A. Andrianingsih, and G. Soepriyono, "Rancang Bangun Website Pengamanan Database E-Voting dengan Menerapkan Algoritma Rivest Shamir Adleman (RSA)," *J. Teknol. Inform. dan Komput.*, vol. 9, no. 2, pp. 1341–1355, 2023, doi: 10.37012/jtik.v9i2.1687.

[14] R. Munir, "Algoritma RSA," vol. 1, no. m, pp. 1–6, 2023.

[15] N. D. Anggoro, C. Suhery, and I. Ruslianto, "Penerapan Algoritma Knapsack dan Fungsi Hash pada Sistem E-Voting (Studi Kasus: Pemilihan Raya Mahasiswa Universitas Tanjungpura Pontianak)," *J. Coding*, vol. 07, no. 01, pp. 85–96, 2019.

[16] A. A. Putra, "Analisis Dan Evaluasi Keamanan Wireless LAN Pada PT. Bumi Jage Dalam," *Proceeding Semin. Nas. Ilmu Komput.*, vol. 1, no. 1, pp. 138–150, 2021, [Online]. Available: https://proceeding.unived.ac.id/index.php/snasikom/article/view/59

[17] A. E. Susanto, "Aplikasi Keamanan Pesan Teks Secara Enkripsi Dan DekripsiMenggunakan Algoritma Rivest Shamir Adleman," *Teknologipintar.org*, vol. 2, no. 8, pp. 1–11, 2022.

[18] H. Hasbulloh, I. Fitri, and S. Ningsih, "Algoritma RSA (Rivest–Shamir–Adleman) pada Sistem Informasi Pemilihan Ketua Organisasi Ikatan Pondok Pesantren Smart-SIPKOTREN," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 3, pp. 424–428, 2022, doi: 10.35870/jtik.v6i3.453.

[19] K. Andriani and B. H. Hayadi, "Pengamanan Data Penjualan Dengan Kriptografi Algoritma Rivest Shamir Adleman (Rsa) Pad a Toko Baju Family," *J. Sci. Soc. Res.*, vol. 5, no. 3, p. 664, 2022, doi: 10.54314/jssr.v5i3.1018.

[20] F. Farhan and D. Leman, "Implementasi Metode Rivest Shamir Adleman (RSA) Untuk Kerahasiaan Database Perum Bulog Kanwil SUMUT," *J. Mach. Learn. Data Anal.*, vol. 2, no. 1, pp. 18–27, 2023, [Online]. Available: https://journal.fkpt.org/index.php/malda/article/view/483/285