Rekayasa Hijau: Jurnal Teknologi Ramah Lingkungan Volume 10 | Nomor 1
ISSN [e]: 2579-4264 | DOI: https://doi.org/10.26760/jrh.V10i1.96-106 Maret 2026

OTP Code Comparison Using RSA and Elgamal Algorithms to Enhance Authentication Security

Asep Rizal Nurjaman[1], Mahardhitya Pratama Wibowo[1]

[1]Program Studi Sistem Informasi, Institut Teknologi Nasional, Bandung, Indonesia

Email: aseprizal@itenas.ac.id

25 Januari 2026 | Revised 2 Februari 2026 | Accepted 10 Februari 2026

ABSTRAK

Jumlah aplikasi yang membutuhkan proses autentikasi yang aman meningkat sebagai akibat dari pertumbuhan pesat teknologi digital. One Time Password (OTP), yang hanya berlaku sekali dan memiliki batas waktu tertentu, adalah salah satu metode autentikasi yang paling umum digunakan. Menjaga kerahasiaan data dan identitas pengguna masih menjadi tantangan karena kerentanan OTP untuk penyadapan atau akses tidak sah. Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja algoritma kriptografi asimetris, khususnya algoritma RSA dan ElGamal, dalam mengamankan kode OTP. Metode penelitian meliputi studi literatur, perancangan algoritma, implementasi algoritma menggunakan bahasa python, dan pengujian performa enkripsi dan dekripsi berdasarkan waktu pada 1 kunci publik dan kunci privat. Hasil pengujian menunjukkan bahwa algoritma ElGamal lebih cocok untuk sistem OTP yang membutuhkan kecepatan dan stabilitas tinggi dalam proses autentikasi. Dengan waktu enkripsi rata-rata 0,003078 ms dan dekripsi rata-rata 0,095154 ms, algoritma ElGamal lebih stabil daripada RSA.

***Kata kunci**: RSA, ELGAMAL, Kriptografi, Kode OTP, Keamanan Autentikasi*

ABSTRACT

The number of applications requiring secure authentication processes is increasing as a result of the rapid growth of digital technology. One-Time Password (OTP), which is valid only once and has a specific time limit, is one of the most commonly used authentication methods. Maintaining the confidentiality of user data and identity remains a challenge due to the vulnerability of OTPs to interception or unauthorized access. This research aims to analyze and compare the performance of asymmetric cryptography algorithms, specifically RSA and ElGamal, in securing OTP codes. The research methods include literature study, algorithm design, algorithm implementation using Python, and performance testing of encryption and decryption based on time for both public and private keys. The test results show that the ElGamal algorithm is more suitable for OTP systems that require high speed and stability in the authentication process. With an average encryption time of 0.003078 milliseconds and an average decryption time of 0.095154 milliseconds, the ElGamal algorithm is more stable than RSA.

***Keywords**: RSA, Elgamal, Cryptography, OTP Code, Authentication*

1. PENDAHULUAN

Perkembangan pesat teknologi digital dan layanan daring membuat mekanisme autentikasi menjadi faktor krusial dalam menjaga kerahasiaan data pengguna. Salah satu metode autentikasi yang paling umum adalah penggunaan One Time Password (OTP), yaitu kode verifikasi sekali pakai yang dikirimkan kepada pengguna dan hanya berlaku dalam jangka waktu tertentu. Meskipun OTP meningkatkan keamanan dibandingkan kata sandi statis, OTP tetap rentan terhadap pencurian atau penyadapan jika tidak dilengkapi mekanisme pengamanan tambahan. kode OTP ini hanya berlaku sekali dan berbatas waktu. Keamanan menjadi fokus utama agar hanya pengguna yang sah yang bisa masuk ke dalam sebuah sistem. Salah satu Salah satu teknik untuk melakukan pengamanan data yaitu kriptografi. Kriptografi adalah ilmu yang mempelajari tentang teknik mengamankan data, baik data yang dikirim atau data yang disimpan. William stalling dalam bukunya menyampaikan jika kriptografi adalah sebuah seni untuk menyandikan pesan [1].

Pada tahun 2022 penelitian terkait algoritma RSA dilakukan untuk mengamankan *file* rekam medis dengan berapa format *file* seperti *doc, *docx, *pdf, *xlsx di sebuah puskesmas, algoritma diimplementasikan dalam bentuk *website*, menyimpulkan bahwa algoritma RSA berhasil diimplementasikan dalam website tersebut, pengujian yang dilakukan adalah menghitung waktu enkripsi dekripsi dan melihat ukuran file setelah dilakukan enkripsi, dimana pada penelitian ini ukuran *file* hasil enkripsi meningkat 3 sampai 6 kali ukuran *file* awal [14]. Penelitian lainnya terkait algoritma RSA di tahun 2023 dilakukan dengan mengimplementasikan algoritma RSA dan algoritma AES serta pada *prototype* aplikasi desktop dalam mengamankan pesan yang dikirim melalui *e-mail*. Penelitian ini memperlihatkan proses yang dilakukan dari awal pembangkitan kunci, enkripsi pesan dan dekripsi pesan. Dalam penelitian ini tidak ada proses pengujian aplikasi atau algoritma yang dikembangkan [4]. Tahun 2024 penelitian terkait algoritma RSA dengan kombinasi algoritma SHA3-512 dilakukan untuk mempertahankan keaslian dokumen. Pada penelitian ini dilakukan pengujian pada skema tanda tangan digital dengan penyerangan *MitM* terhadap skema yang dibuat dan menghasilkan perubahan ukuran *file* setelah dilakukan penanda tanganan digital dengan perubahan ukuran file rata-rata 0.3 MB [2]. Sementara penelitian terkait algoritma ElGamal dilakukan pada tahun 2021 menyampaikan bahwa beberapa modifikasi dilakukan pada ElGamal Kriptosistem tradisional untuk mengurangi ukuran *ciphertext* dan mempercepat waktu eksekusi. Skema yang diusulkan mengurangi tingkat ekspansi dalam Elgamal Kriptosistem tradisional hingga 89%. Selain itu, mempercepat waktu eksekusi yang membuat skema yang diusulkan berkinerja lebih baik daripada ElGamal Kriptosistem tradisional [3]. Penelitian lain di tahun 2023, algoritma ElGamal yang dikombinasikan dengan LSB pada penyisipan pesan teks terenkripsi berhasil diimplementasikan dimana pengujian pada penelitian ini hanya menggunakan *blackbox testing* dengan memperhatikan hasil *input* dan *output* saja [13].

Penelitian ini dilakukan untuk menjawab kesenjangan tersebut dengan membandingkan dua algoritma asimetris yang paling banyak digunakan, yaitu RSA dan ElGamal, pada proses enkripsi dan dekripsi kode OTP. RSA dipilih karena merupakan algoritma kriptografi publik paling klasik dan banyak digunakan dalam berbagai aplikasi keamanan, sedangkan ElGamal dipilih karena memiliki dasar keamanan logaritma diskrit dan dikenal lebih efisien di beberapa skenario. Dengan membandingkan keduanya secara langsung pada konteks OTP, penelitian ini diharapkan dapat memberikan dasar empiris dalam memilih algoritma kriptografi yang lebih tepat untuk meningkatkan keamanan autentikasi berbasis OTP.

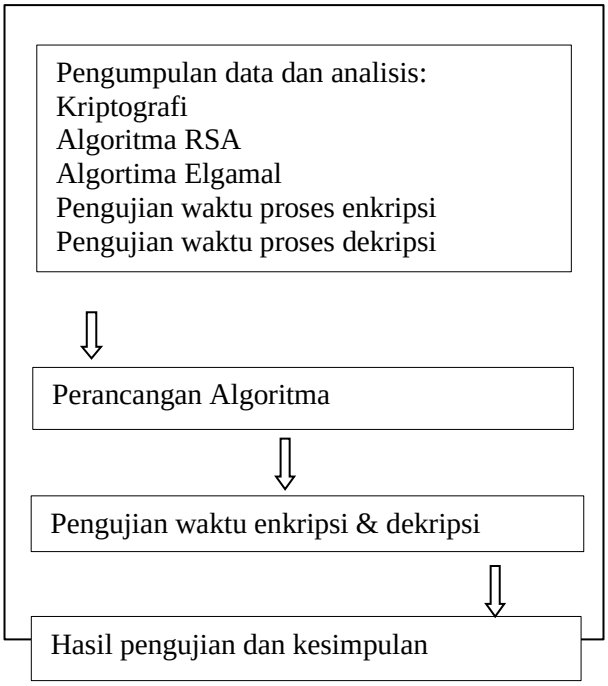
Oleh karena itu, penelitian ini dilaksanakan untuk menganalisis dan membandingkan kinerja algoritma RSA dan ElGamal pada proses enkripsi dan dekripsi kode OTP. Dengan adanya pembandingan ini diharapkan dapat diperoleh gambaran empiris mengenai algoritma kriptografi asimetris mana yang lebih

sesuai dan lebih stabil digunakan pada sistem autentikasi berbasis OTP, sehingga dapat menjadi acuan dalam pengembangan aplikasi keamanan di masa mendatang.

2. METODOLOGI

2.1 Metodologi Penelitian

Pada penelitian ini dilakukan beberapa langkah proses dimulai dari pengumpulan data dan analisis sampai dengan proses akhir berupa pengujian dan kesimpulan. Proses penelitian dapat dilihat pada Gambar 1.

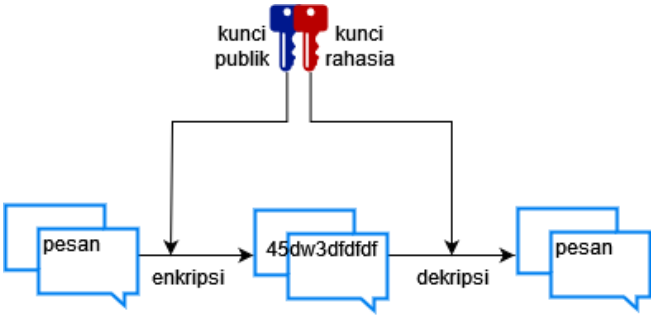


Gambar 1. Alur proses penelitian

Tahap pertama pada metodologi penelitian ini adalah fase pengumpulan data dan analisis, dimana terdapat beberapa kajian teori yang dibahas, yaitu :

a. Kriptografi (Skema Asimetris)

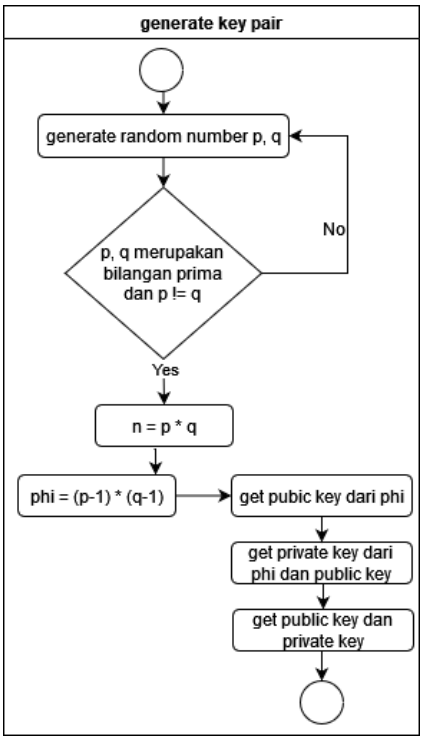
   Kriptografi merupakan bidang keilmuan dalam mengamankan data / pesan. Pada umumnya kriptografi membuat pesan atau data menjadi tidak terbaca dengan menggunakan teknik enkripsi dan data yang terlihat acak itu bisa di kembalikan ke data atau pesan semula dengan teknik dekripsi [16]. Pada penelitian ini, kriptografi digunakan untuk mengacak kode OTP. Pada penelitian ini, jenis kriptografi yang digunakan adalah kriptografi asimetris dimana proses enkripsi menggunakan kunci publik dan dekripsi menggunakan kunci rahasia (menggunakan kunci yang berbeda untuk enkripsi dan dekripsi). Gambar 2 merupakan skema proses enkripsi dan proses dekripsi dengan skema asimetris.



Gambar 2. Skema proses enkripsi dan dekripsi asimetris pada kriptografi

b. Algoritma RSA

Algoritma RSA merupakan algoritma asimetris dimana dibutuhkan kunci publik dan kunci rahasia untuk melakukan enkripsi / dekripsi. Pada penelitian ini, algoritma RSA digunakan untuk mengenkripsi pesan sebelum pesannya disisipkan dalam sebuah gambar. Algoritma RSA merupakan algoritma asimetris yang digunakan untuk mengamankan pesan yang digunakan dalam penelitian ini yaitu algoritma RSA. Proses pembangkitan pasangan kunci pada RSA ditampilkan pada Gambar 3.



Gambar 3. Proses pembangkitan kunci pada RSA [17]

Langkah untuk membuat pasangan kunci pada algoritma RSA yaitu ambil dua buah bilangan prima sembarang, p dan q, lalu hitung r dari p*q dimana p tidak sama dengan q. Hitung *phi*(r) dengan mengalikan (p - 1) dan (q - 1). Pilihlah kunci publik (PK) yang relatif prima terhadap *phi*(r). Untuk kunci rahasia (SK) didapat dari $(1 + m\Phi (r)) / PK$ [17]. Untuk melakukan proses enkripsi pada algoritma RSA, maka ubah terlebih dahulu karakter ke dalam ASCII. Selanjutnya proses enkripsi dilakukan per karakter dengan dengan perhitungan rsa yaitu $y_i = x_i^{PK}$ *mod* r dimana x_i merupakan *index* dari karakter pada pesan yang telah diubah ke dalam ASCII. Sementara untuk melakukan

dekripsi pada algoritma RSA menggunakan formula $x_i = y_i^{SK}$ *mod* r dimana y_i merupakan indeks dari karakter pada *ciphertext* yang akan didekripsi.

c. Algortima ElGamal

ElGamal merupakan algoritma kriptografi asimetris yang keamanannya bergantung pada tingkat kesulitan masalah logaritma diskrit [18]. Algoritma ElGamal terdiri dari tiga fase:

1. Algoritma pembangkitan pasangan kunci
   - Ambil bilangan prima besar (p).
   - Pilih bilangan sebagai generator (g) dengan ketentuan $1 < g < p - 1$.
   - Pilih bilangan bulat (x) sehingga $1 < x < p - 2$, x dimana kunci privat.
   - Hitung $y = g^x$ *mod* p.
   - Informasi kunci publik = (p, g, y), Kunci privat = x.
2. Algoritma enkripsi

   Proses enkripsi dilakukan dengan menggunakan informasi kunci publik. Langkah-langkah untuk mengenkripsi pesan rahasia:
   - Pengirim hanya menerima informasi kunci publik, yang akan memungkinkannya untuk mengenkripsi.
   - Pengirim mengodekan pesan (m) dengan mengubah representasi *string* menjadi nilai numerik yang sesuai.
   - Pengirim memilih bilangan bulat (k) sehingga $1 < k < p - 2$.
   - Pengirim menghitung $y = g^k$ *mod* p.
   - Pengirim juga menghitung $z = (y^k \times m)$ *mod* p.
   - Pengirim kemudian mengirimkan informasi *ciphertext* C = (y, z) ke penerima.
3. Algoritma dekripsi

   Proses dekripsi dilakukan dengan menggunakan informasi kunci privat. Langkah-langkah untuk mendekripsi *ciphertext*:
   - Penerima membutuhkan kunci pribadi (x) untuk menyelesaikan proses dekripsi.
   - Penerima mengambil informasi *ciphertext* C = (y, z). Kemudian ia menghitung $r = y^{(p-1-x)}$ *mod* p.
   - Penerima akhirnya menghitung $m = (r \times z)$ *mod* p untuk mengekstrak pesan rahasia.

Tahap kedua pada metodologi penelitian ini adalah perancangan algoritma pada proses pengamanan kode OTP, baik algoritma RSA maupun Algoritma ElGamal. Proses secara umum akan terlihat sama baik pengamanan kode OTP dengan algoritma RSA maupun algoritma ElGamal. Perbedaannya hanya pada proses enkripsi dan dekripsi kode OTP.

Tahap ketiga merupakan pengujian waktu enkripsi dan dekripsi pada kedua algoritma. Dimana pada pengujian waktu enkripsi dan dekripsi akan diambil sampel 100 kali pembuatan kode OTP yang kan diamankan dengan algoritma RSA dan 100 kali pembuatan kode OTP yang akan diamankan dengan skema dengan algoritma ElGamal. Kode OTP dibatasi hanya 8 karakter angka.

Pada tahap terakhir akan dilihat waktu dari proses enkripsi dan dekripsi baik dengan algoritma RSA dan algoritma ElGamal, akan diambil waktu tercepat proses enkripsi dan dekripsi untuk algoritma RSA dan algoritma ElGamal. Sehingga bisa menjadi referensi untuk digunakan pada aplikasi.

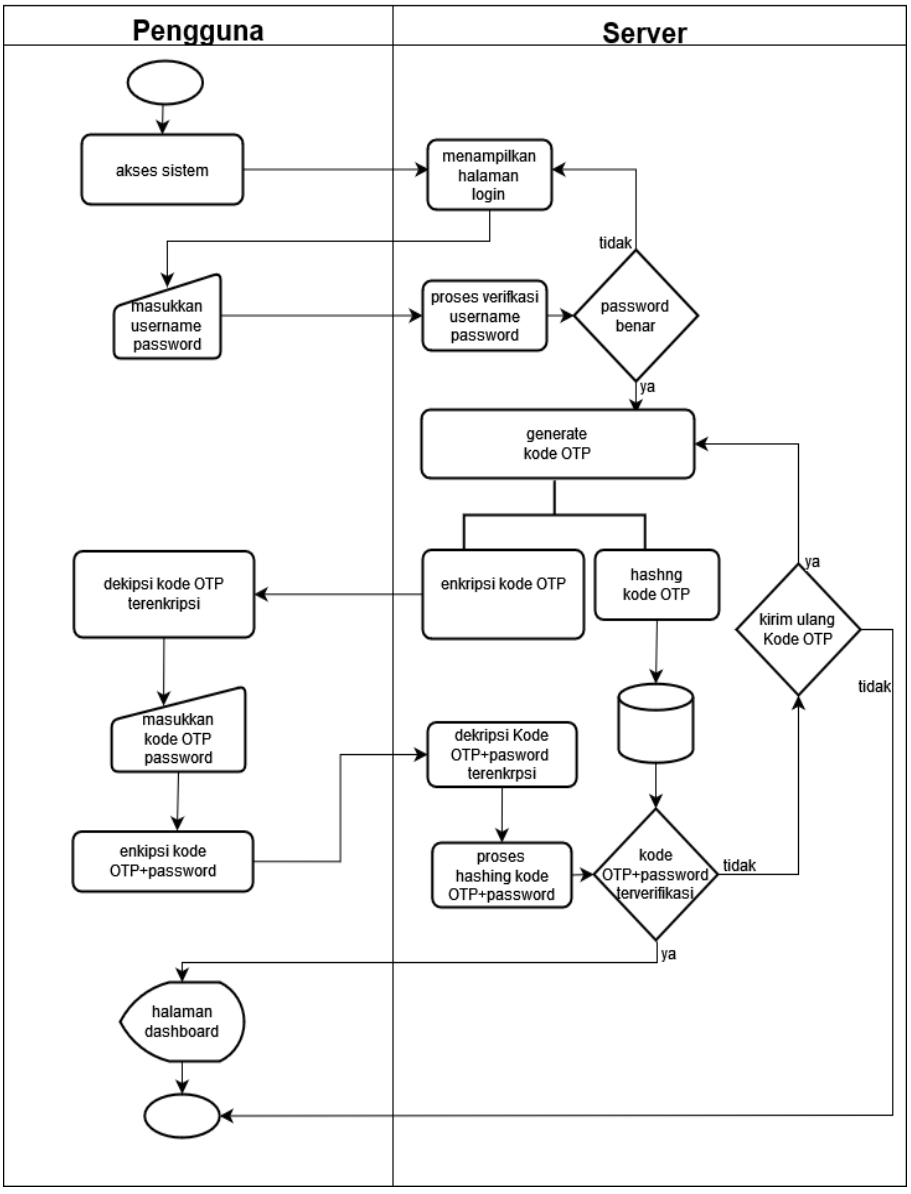
3. HASIL DAN PEMBAHASAN

3.1 Pembahasan

Desain dari algoritma kriptografi yang digunakan untuk mengamankan kode OTP dengan algoritma RSA dan algoritma ElGamal bisa dilihat pada Gambar 4, dimana proses yang berbeda adalah saat proses enkripsi dan dekripsi antara menggunakan algoritma RSA atau ElGamal.

Alat yang digunakan untuk pengujian waktu enkripsi dan dekripsi algoritma RSA dan ElGamal sebuah laptop dengan spesifikasi sebagai berikut :

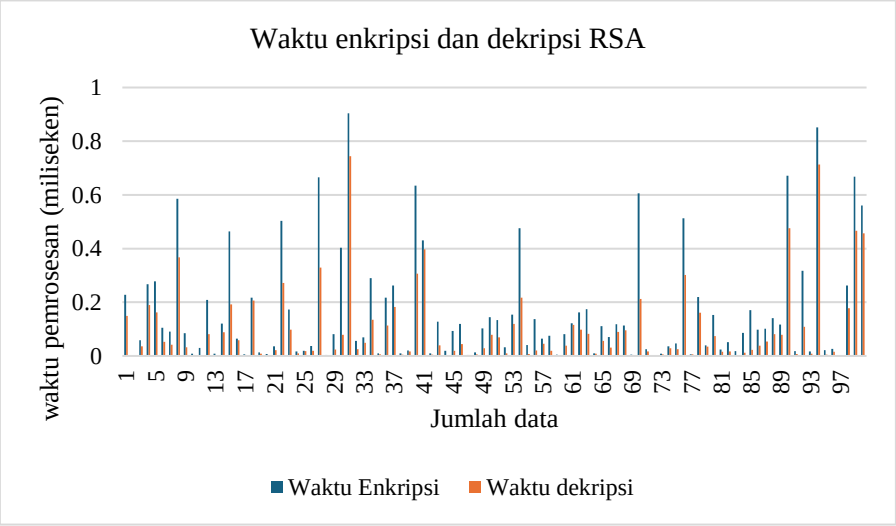
System Manufacturer	: Dell Inc.
System Model	: Latitude E6440
BIOS	: A14 (*type*: UEFI)
Processor	: Intel(R) Core(TM) i7-4610M CPU @ 3.00GHz (4 CPUs), ~3.0GHz
Memory	: 8192 MB RAM



Gambar 4. Proses pembangkitan kode OTP dengan algoritma RSA dan ElGamal

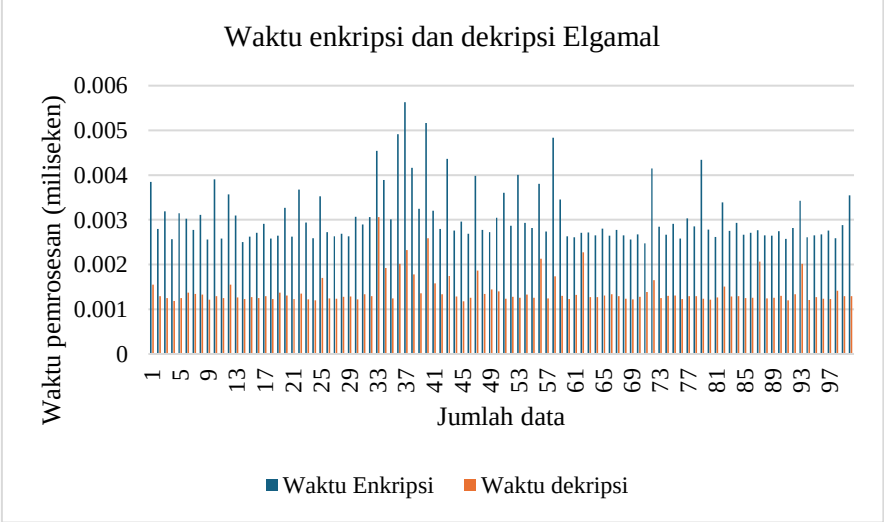
3.2 Hasil

Hasil dari pengujian algoritma RSA dan ElGamal dengan 100 data kode OTP pada proses enkripsi dan dekripsi akan dibandingkan dan dianalisis. Beberapa parameter yang ditinjau di antaranya proses waktu enkripsi, proses waktu dekripsi dari kedua algoritma tersebut dengan mempertimbangkan kecepatan proses dan waktu rata-rata yang relatif cepat. Gambar 5 merupakan proses waktu enkripsi dan dekripsi dengan algoritma RSA yang dilakukan pada 100 kode OTP yang dibangkitkan dengan panjang kode OTP delapan karakter angka. Dari gambar 5 dapat dilihat jika waktu tercepat untuk proses enkripsi dengan algoritma RSA adalah 0.001496 *ms* sementara waktu terlama untuk proses enkripsi adalah 0.903924 *ms* dan waktu tercepat untuk proses dekripsi dengan algoritma RSA adalah 0.000860 *ms* sementara waktu terlama untuk proses dekripsi adalah 0.743633 *ms*. Sehingga rata-rata waktu enkripsi kode OTP dari 100 kali percobaan adalah 0.162804 ms dan rata-rata waktu dekripsi kode OTP dari 100 kali percobaan adalah 0.095154 *ms*.



Gambar 5. Waktu enkripsi dan dekripsi kode OTP dengan algoritma RSA

Gambar 6 merupakan proses waktu enkripsi dan dekripsi dengan algoritma ElGamal yang dilakukan pada 100 kode OTP yang dibangkitkan dengan panjang kode OTP delapan karakter angka.

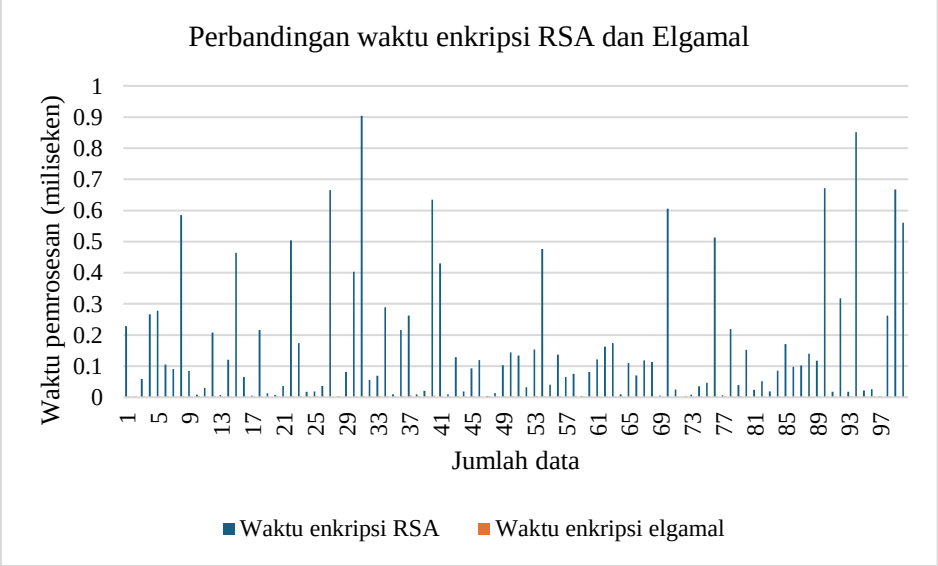


Gambar 6. Waktu enkripsi dan dekripsi kode OTP dengan algoritma ElGamal

Waktu tercepat untuk proses enkripsi dengan algoritma ElGamal adalah 0.002475 ms sementara waktu terlama untuk proses enkripsi adalah 0.005630 ms dan waktu tercepat untuk proses dekripsi dengan algoritma RSA adalah 0.001180 ms sementara waktu terlama untuk proses dekripsi adalah 0.003063 ms. Sehingga rata-rata waktu enkripsi kode OTP dari 100 kali percobaan adalah 0.003078 ms dan rata-rata waktu dekripsi kode OTP dari 100 kali percobaan adalah 0.001405 ms.

3.2.1 Hasil perbandingan waktu enkripsi RSA dan ElGamal

Perbandingan waktu enkripsi dengan algoritma RSA dan ElGamal dapat dilihat pada gambar 7. Dimana waktu maksimum yang dibutuhkan untuk melakukan enkripsi pada algoritma ElGamal lebih kecil dibandingkan waktu yang dibutuhkan algoritma RSA.



Gambar 7. Perbandingan waktu enkripsi RSA dan ElGamal

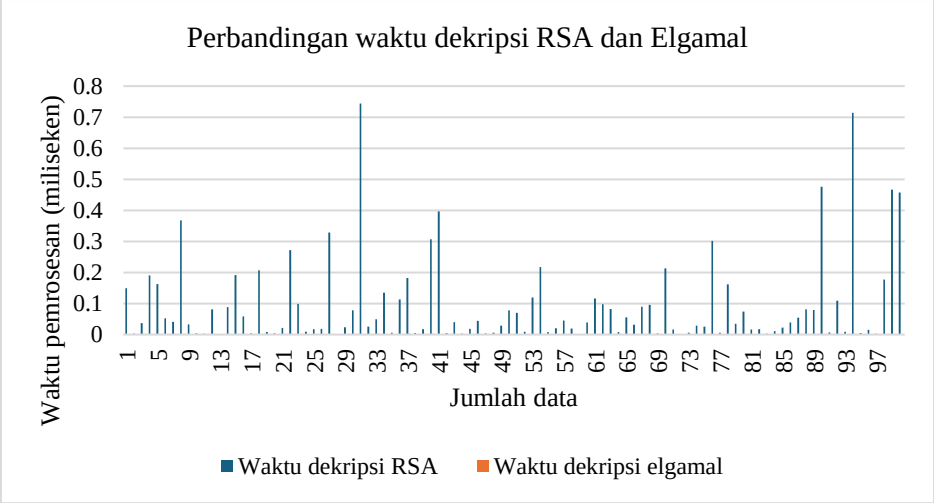
Ringkasan waktu enkripsi tercepat dan terlama untuk algoritma RSA dan ElGamal dari gambar 7 bisa dilihat pada tabel 1. Dari tabel 1 bisa disimpulkan bahwa waktu enkripsi rata-rata pada algoritma ElGamal lebih cepat dibandingkan rata-rata waktu enkripsi pada algoritma RSA.

Tabel 1. Waktu enkripsi Algoritma RSA dan ElGamal

Algoritma	Waktu enkripsi tercepat	Waktu enkripsi terlama	Rata-rata waktu enkripsi
RSA	0.001496 ms	0.903924 ms	0.162804 ms
ElGamal	0.002476 ms	0.005630 ms	0.003078 ms

3.2.2 Hasil perbandingan waktu dekripsi RSA dan ElGamal

Perbandingan waktu dekripsi dengan algoritma RSA dan ElGamal dapat dilihat pada gambar 8, waktu maksimum yang dibutuhkan untuk melakukan dekripsi pada algoritma ElGamal lebih kecil dibandingkan waktu yang dibutuhkan algoritma RSA



Gambar 8. Perbandingan waktu dekripsi RSA dan ElGamal

Ringkasan waktu dekripsi tercepat dan terlama untuk algoritma RSA dan ElGamal dari gambar 8 bisa dilihat pada tabel 2. Dimana pada tabel 2 akan disimpulkan bahwa waktu dekripsi rata-rata pada algoritma ElGamal lebih cepat dibandingkan rata-rata waktu enkripsi pada algoritma RSA.

Tabel 2. Waktu dekripsi Algoritma RSA dan ElGamal

Algoritma	Waktu enkripsi tercepat	Waktu enkripsi terlama	Rata-rata waktu enkripsi
RSA	0.000860 ms	0.743633 ms	0.095154 ms
ElGamal	0.001181 ms	0.003064 ms	0.001405 ms

Dari tabel 1 dan tabel 2 bisa disimpulkan jika waktu proses untuk enkripsi dan dekripsi menggunakan algoritma ElGamal lebih stabil dan waktu rata-rata yang dibutuhkan untuk prosesnya lebih kecil dibandingkan algoritma RSA.

4. KESIMPULAN

Pada penelitian ini, algoritma yang digunakan untuk mengamankan kode OTP adalah RSA dan ElGamal. Waktu enkripsi dan dekripsi dari kedua algoritma tersebut dibandingkan. Kode OTP yang digunakan berjumlah delapan digit angka acak. Secara umum prosesnya yaitu server membangkitkan delapan digit angka acak yang di enkripsi dan dikirim ke penerima. Penerima yang sah dipastikan bisa membuka kode enkripsi dan penerima mengirim kode hasil dekripsi yang kemudian di enkripsi kembali, saat server dapat memvalidasi dan membuka kode enkripsi dari penerima maka server bisa yakin jika data yang dikirim dari penerima yang sah dan untuk server sehingga menghasilkan proses *mutual* autentikasi. Dari proses pengujian algoritma, algoritma ElGamal membutuhkan waktu yang stabil dari 100 kali percobaan pada proses enkripsi dan dekripsi dimana rata-rata waktu yang dibutuhkan algoritma ElGamal untuk melakukan enkripsi yaitu 0.003078 *ms*, lebih cepat dibandingkan rata-rata waktu enkripsi RSA yaitu 0.162804 *ms*. Begitu juga saat proses dekripsi dimana waktu rata-rata yang dibutuhkan algoritma ElGamal yaitu 0.001405, sementara waktu rata-rata dekripsi RSA 0.095154. Untuk penelitian selanjutnya diharapkan algoritma ini bisa diimplementasikan pada aplikasi desktop, *web base* dan *mobile*.

DAFTAR PUSTAKA

[1] W. Catur, U. Putri, R. Marwati, dan S. M. Gozali, "Penggabungan Kriptografi Rivest Shamir Adleman (Rsa) dan Advanced Encryption Standard (Aes) Pada Aplikasi Pengirim E-Mail," vol. 3, no. 2, hlm. 92–101, 2023.

[2] O. F. Abdelwahab, A. I. Hussein, H. F. A. Hamed, H. M. Kelash, dan A. A. M. Khalaf, "Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques to Hide Data," dalam *Procedia Computer Science*, Elsevier B.V., 2021, hlm. 5–12. doi: 10.1016/j.procs.2021.02.002.

[3] A. Cahya Putra dan M. Simanjuntak, "Penerapan Algoritma Rivest Shamir Adleman (RSA) untuk Mengamankan Database Program Keluarga Harapan (PKH)," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 5, no. 1, 2021.

[4] I. Suhendra dan A. Maslan, "Analisis dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi Secure Hash Algorithm," *Jurnal Comasie*, vol. 10, no. 05, 2024.

[5] T. Mahesti, A. F. Ciptaningtyas, A. Astungkara, J. A. Politeknik, dan N. Semarang, "Perbandingan Penggunaan Algoritma Kriptografi DES, RSA, Modifikasi DES dan Modifikasi RSA untuk Penyandian Database."

[6] J. Felisha, "Analisis Perbandingan Algoritma RSA dengan ElGamal pada Tanda Tangan Digital."

[7] A. M. Fajrin, J. R. Benedict, dan H. J. Kusuma, "Analisis Performa dari Algoritma Kriptografi RSA dan ElGamal dalam Enkripsi dan Dekripsi Pesan," vol. 8, hlm. 91–98, [Daring]. Tersedia pada: https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

[8] Y. Pratama Putra, F. Nuraeni, R. Ajji Jatnika, P. Studi Teknik Informatika, dan S. Tasikmalaya, "Implementasi Kriptografi Dalam Pengamanan Database E-Voting Menggunakan Algoritma Rsa Dan Base64 Berbasis Progresive Web Apps (Studi Kasus: Pemilihan Presiden Mahasiswa STMIK Tasikmalaya)," 2021.

[9] F. Farhan dan D. Leman, "Implementasi Metode Rivest Shamir Adleman (RSA) Untuk Kerahasiaan Database Perum Bulog Kanwil SUMUT," *Journal of Machine Learning and Data Analytics (MALDA)*, vol. 02, no. 01, hlm. 18–27.

[10] E. Saragih, D. Siregar, dan H. Dafitri, "Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer) Implementasi Penyisipan Pesan Teks Terenkripsi Menggunakan Kriptografi ElGamal pada Citra Digital Menggunakan Steganogarafi LSB", [Daring]. Tersedia pada: https://ojs.trigunadharma.ac.id/index.php/jis/index

[11] R. Davia, A. Huday, dan S. Waluyo, "Pengamanan File Rekam Medis pada Puskesmas Larangan Utara Menggunakan Algoritma Kriptografi RSA Berbasis Web," 2022.

[12] "SHA-3 standard :," 2015. doi: 10.6028/NIST.FIPS.202.

[13] J. Kelsey, S. Change, dan R. Perlner, "SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash," Gaithersburg, MD, Des 2016. doi: 10.6028/NIST.SP.800-185.

[14] A. Hermawan, E. Iman, dan H. Ujianto, "Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA," vol. 5, no. 2, 2021, doi: 10.30743/infotekjar.v5i2.3585.

[15] I. A. Darmawan, "Kriptografi Algoritma RSA untuk Pengamanan Database Berbasis Java Dekstop pada SMA Muhammadiyah 15 Jakarta Barat," 2018.

[16] W. Stalling, *Cryptography and Network Security: Principles and Practice*. New Jersey: Prentice Hall Press, 2013.

[17] A. Rizal Nurjaman, "Penanda Tanganan Dokumen Digital Pada Sistem Penyimpanan File Menggunakan Kombinasi Algoritma SHA3-512 dan RSA untuk Mempertahankan Keaslian Data Dokumen," 2024.

[18] H. I. Hussein dan W. M. Abduallah, "An efficient ElGamal cryptosystem scheme," *International Journal of Computers and Applications*, vol. 43, no. 10, hlm. 1088–1094, 2021, doi: 10.1080/1206212X.2019.1678799.