E-ISSN : 2809-3755
P-ISSN: 2809-3526

INTERVAL: Jurnal Ilmiah Matematika
Volume 3, No. 2, September 2023, pp. 92-101

PENGGABUNGAN KRIPTOGRAFI RIVEST SHAMIR ADLEMAN (RSA) DAN ADVANCED ENCRYPTION STANDARD (AES) PADA APLIKASI PENGIRIM E-MAIL

Widya Catur Utami Putri[1], Rini Marwati[2], Sumanang Muhtar Gozali[3]
[1,2,3]Program Studi Matematika, FPMIPA, Universitas Pendidikan Indonesia
e-mail: widyacatur@upi.edu

Diterima: 3 Agustus 2023 , disetujui: 9 Agustus 2023, dipublikasi: 31 September 2023

Abstract**:** *One of the most commonly used communication tools is e-mail. Messages sent via e-mail can contain confidential information, security measures are needed for confidential messages sent via e-mail. Cryptography is one of the security measures that can be taken. Based on the types of keys, there are two types of cryptography namely symmetric and asymmetric. Symmetric cryptography uses the same key in the encryption and decryption process, while asymmetric cryptography uses different keys. Symmetric keys can be encrypted by asymmetric keys to make the transmission of secret keys more secure. The symmetric and asymmetric cryptography chosen are AES (Advanced Encryption Standard) and enhanced RSA (Rivest Shamir Adleman). RSA cryptography is known for its strong security based on the difficulty of exponential and factorization operations, while AES cryptography is a secure cryptography standard that has a relatively fast computation time. An e-mail application to send secret messages using combined cryptography algorithm constructed using Python programming language.*

Keywords*: cryptography, rsa, advanced encryption standard, e-mail*

Abstrak**:** *Salah satu sarana komunikasi yang paling umum digunakan adalah e-mail. Pesan yang dikirim melalui e-mail dapat berisi informasi rahasia, tindakan pengamanan diperlukan untuk pesan rahasia yang dikirim melalui e-mail. Kriptografi merupakan salah satu langkah pengamanan yang dapat dilakukan. Berdasarkan jenis kuncinya, kriptografi dibedakan menjadi dua jenis yaitu simetris dan asimetris. Kriptografi simetris menggunakan kunci yang sama dalam proses enkripsi dan dekripsi, sedangkan kriptografi asimetris menggunakan kunci yang berbeda. Kunci simetris dapat dienkripsi dengan kunci asimetris untuk membuat transmisi kunci rahasia lebih aman. Kriptografi simetris dan asimetris yang dipilih adalah AES (Advanced Encryption Standard) dan Enhanced RSA (Rivest Shamir Adleman). Kriptografi RSA dikenal dengan keamanannya yang kuat berdasarkan kesulitan operasi eksponensial dan faktorisasi, sedangkan kriptografi AES merupakan standar kriptografi aman yang memiliki waktu komputasi relatif cepat. Sebuah aplikasi e-mail untuk mengirim pesan rahasia menggunakan algoritma kriptografi gabungan yang dibangun menggunakan bahasa pemrograman Python.*

Kata Kunci*: kriptografi, rsa, advanced encryption standard, e-mail*

PENDAHULUAN

Salah satu sarana komunikasi yang paling umum digunakan adalah surat elektronik atau *e-mail*. Pesan-pesan yang dikirim melalui *e-mail* dapat berupa informasi rahasia. Jika pesan-pesan ini jatuh ke tangan yang salah dapat mengakibatkan berbagai kerugian, contohnya pencurian data atau penyalahgunaan identitas. Tercatat sejumlah serangan *cyber* yang menargetkan *e-mail* dalam beberapa tahun terakhir, contohnya *Gmail Phishing Attack* di tahun 2017 dan *Hyperscrape* di tahun 2021. Dalam kehidupan sehari-hari pun, pengguna *e-mail* terkadang menyetujui kebijakan privasi tanpa membaca secara detail ketentuan-ketentuannya. Oleh karena itu, untuk mengirim pesan-pesan rahasia melalui *e-mail* diperlukan tindakan pengamanan. Salah satunya melalui penerapan kriptografi.

Dalam kriptografi, pesan yang dapat dimengerti (plainteks) diubah menjadi pesan tersamar (*cipherteks*) agar pembaca yang tidak berhak tidak dapat memahami makna pesan tersebut. Kriptografi simetris menggunakan kunci yang sama dalam proses enkripsi dan dekripsi, sedangkan kriptografi asimetris menggunakan kunci yang berbeda, yang terdiri dari kunci publik yang bersifat tidak rahasia dan kunci *privat* yang bersifat rahasia [1]. Dalam kriptografi simetris, permasalahan muncul ketika kunci rahasia perlu ditransmisikan secara aman. Kunci simetris ini kemudian dapat dienkripsi dengan kunci asimetris untuk membuat transmisi kunci rahasia lebih aman. Skema ini termasuk ke dalam kriptografi hibrida [10]. Dalam hal ini, algoritma kriptografi yang dipilih adalah algoritma RSA yang ditingkatkan (asimetris) dan algoritma AES (simetris).

Algoritma Rivest Shamir Adleman (RSA) merupakan salah satu skema kriptografi asimetris. RSA terkenal akan keamanannya yang kuat didasarkan pada sulitnya operasi eksponensial dan masalah pemfaktoran bilangan bulat [2]. Algoritma kriptografi RSA kemudian ditingkatkan agar kemungkinan kriptanalisis dapat diminimalisir. *Advanced Encryption Standard* (AES) merupakan standar kriptografi pengganti *Data Encryption Standard (DES)* yang dihasilkan dari sayembara terbuka oleh *National Institute of Standards and Technology (NIST).*

Terdapat beberapa penelitian sebelumnya yang mengkaji RSA dan AES. Fatma [3] mengimplementasikan AES dan LSB untuk digunakan enkripsi gambar dan menghasilkan waktu yang cepat yaitu hanya dalam kisaran waktu 100 milidetik. Namun, Bimantoro [4] mengungkapkan bahwa penggunaan AES saja dinilai kurang karena harus mengirimkan kunci bersamaan dengan pesan enkripsi. Hermawan [5] menggunakan algoritma RSA dan AES pada skema kriptografinya. Dalam penelitian tersebut belum diterapkan algoritma RSA yang ditingkatkan dan penerapannya pada aplikasi *e-mail*. Pada algoritmanya pula, terdapat tahapan transmisi kunci *privat* yang seharusnya menjadi rahasia. Bimantoro [4] menyimpulkan bahwa gabungan metode kriptografi RSA dan AES menghasilkan performa yang baik dan dapat memproses data dalam jumlah besar dalam waktu milidetik. Namun, disebutkan pula bahwa dalam pengujiannya desain yang diajukan masih belum sempurna dan dapat ditingkatkan lagi performa dan keamanannya.

Dalam penelitian ini, digunakan algoritma kriptografi RSA ditingkatkan untuk menimalisir kemungkinan kriptanalisis, kemudian kunci rahasia dienkripsi oleh kunci dari algoritma kriptografi simetris agar transmisi kunci rahasia dapat dilakukan secara aman dan menyulitkan pihak ketiga untuk mendapatkan informasi rahasia yang dikirimkan melalui skema kriptografi simetris.

METODOLOGI PENELITIAN

Kriptografi Rivest Shamir Adleman (RSA) yang Ditingkatkan

Algoritma RSA dibuat oleh Ron Rivest, Adi Shamir, dan Leonard Adleman, tiga orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976. RSA termasuk ke dalam skema kriptografi asimetris. Kriptografi asimetris merupakan algoritma kriptografi yang menggunakan kunci yang berbeda yaitu kunci publik (*public key*) dan kunci rahasia (*private key*) pada proses enkripsi dan dekripsinya [6]. Kunci publik dapat mengenkripsi pesan namun tidak dapat mendekripsi pesan.

Misalnya, Bob ingin mengirim pesan kepada Alice menggunakan kriptografi asimetris, hal yang harus dilakukan oleh Bob dan Alice adalah:

1. Bob membangkitkan kunci publik dan kunci *privat*.
2. Bob memberikan kunci publik kepada Alice.
3. Alice mengenkripsi pesan menggunakan kunci publik dari Bob, kemudian mengirimkan pesan yang terenkripsi kepada Bob.
4. Bob mendekripsi pesan menggunakan kunci *privat* yang dimilikinya.

Skema kriptografi asimetris ini dapat mengatasi masalah keamanan transmisi kunci rahasia.

Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran ini dilakukan untuk memperoleh kunci *privat* yang dapat mendekripsi pesan. Oleh karena itu, RSA dianggap aman. Namun, RSA juga memiliki kelemahan. Proses enkripsi dan dekripsi dalam RSA cenderung memakan waktu yang lama, terutama ketika digunakan untuk mengenkripsi atau mendekripsi data dengan ukuran yang besar. Oleh karena itu, RSA umumnya digunakan untuk pertukaran kunci dan pengamanan data yang relatif kecil, sedangkan algoritma simetris seperti AES digunakan untuk mengenkripsi data dalam skala besar. Properti algoritma RSA sesuai dengan yang tercantum pada Tabel 1.

Tabel 1. **Properti Algoritma RSA**

Properti	Properti Algoritma	Kerahasiaan
p	bilangan prima	rahasia
q	bilangan prima	rahasia
n	$n = p \times q$	tidak rahasia
$\phi(n)$	$\phi(n) = (p - 1)(q - 1)$	rahasia
e	kunci enkripsi, yang memenuhi $FPB(e, \phi(n)) = 1$	tidak rahasia

Pada tahap pembangkitan kunci, dilakukan beberapa langkah sebagai berikut.

1. Pilih dua bilangan prima p, dan q.
2. Hitung
$$n = p \times q \quad (1)$$
($p \neq q$ agar n tidak mudah difaktorkan).
3. Hitung
$$\phi(n) = (p - 1)(q - 1) \quad (2)$$
4. Pilih kunci publik e, yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan d yang memenuhi

$$e.\, d = 1\ mod\ \phi(n) \tag{3}$$

Kemudian, proses enkripsinya adalah sebagai berikut.

1. Ambil kunci publik penerima pesan e, dan modulus n.
2. Nyatakan *plainteks* m menjadi blok-blok m_1, m_2, …, sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang $[0, n-1]$.
3. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus

$$c_i = {m_i}^e\ mod\ n \tag{4}$$

Pada proses dekripsi, setiap blok *cipherteks* c_i didekripsi kembali menjadi blok m_i dengan rumus ${m_i} = {c_i}^d\ mod\ n$.

Perbedaan algoritma RSA standar dengan algoritma RSA yang ditingkatkan terdapat pada proses pembangkitkan kunci. Proses pembangkitan kunci, enkripsi, dan dekripsi menggunakan penggabungan algoritma RSA yang ditingkatkan dan AES adalah sebagai berikut [2].

1. Pilih tiga bilangan prima p_1, p_2, dan p_3.
2. Hitung

$$n = p_1 \times p_2 \times p_3 \tag{5}$$

dengan $p_i \neq p_j$, $i \neq j$ agar n tidak mudah difaktorkan

3. Hitung

$$\phi(n) = (p_1 - 1) \times (p_2 - 1) \times (p_3 - 1) \tag{6}$$

4. Pilih kunci publik e, yang relatif prima terhadap $\phi(n)$
5. Bangkitkan d yang memenuhi

$$e.\, d = 1\ mod\ \phi(n) \tag{7}$$

Kriptografi *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES), yang dibuat dan diimplementasikan oleh Joan Daemen dan Vincent Rijmen, adalah sistem enkripsi data yang ditetapkan oleh *U.S National Institute of Standards and Technology* (NIST) pada tahun 2001. AES banyak digunakan saat ini karena jauh lebih kuat daripada DES dan *triple* DES. AES termasuk kepada jenis algoritma kriptografi yang bersifat simetri dan *block cipher*.

Kriptografi simetris merupakan skema kriptografi yang hanya menggunakan satu kunci atau kunci yang sama untuk proses enkripsi dan dekripsinya [6]. Dalam penerapan skema kriptografi simetris, selain mengirimkan *cipherteks*, pengirim pesan juga harus mengirimkan kunci rahasia agar penerima dapat mendekripsi *cipherteks* tersebut. Dalam *cipher* blok, *plainteks*/*cipherteks* diproses dalam bentuk blok-blok bit atau *bytes*. Rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya. Algoritma AES memiliki 3 jenis panjang kunci kriptografi yang berbeda yaitu 128, 192, dan 256 bit. Untuk ukuran blok selalu sama, yaitu 128-bit.

Blok-blok data masukan dan kunci dioperasikan dalam bentuk *array*. Operasi AES dilakukan terhadap *array of byte* yang disebut dengan *state*. Setiap *state* akan mengalami proses yang secara garis besar terdiri dari empat tahap [7, 8], yaitu:

1. *Add Round Key* (Transformasi Penambahan Kunci)

   Dalam proses ini *subkey* digabungkan dengan *state*. Proses penggabungan ini menggunakan operasi XOR untuk setiap *byte* dari *subkey* dengan *byte* yang bersangkutan dari *state*. Untuk setiap tahap, *subkey* dibangkitkan dari kunci utama dengan menggunakan proses *key schedule*. Setiap *subkey* berukuran sama dengan *state* yang bersangkutan.
2. *SubBytes* (Transformasi Substitusi *Byte*)

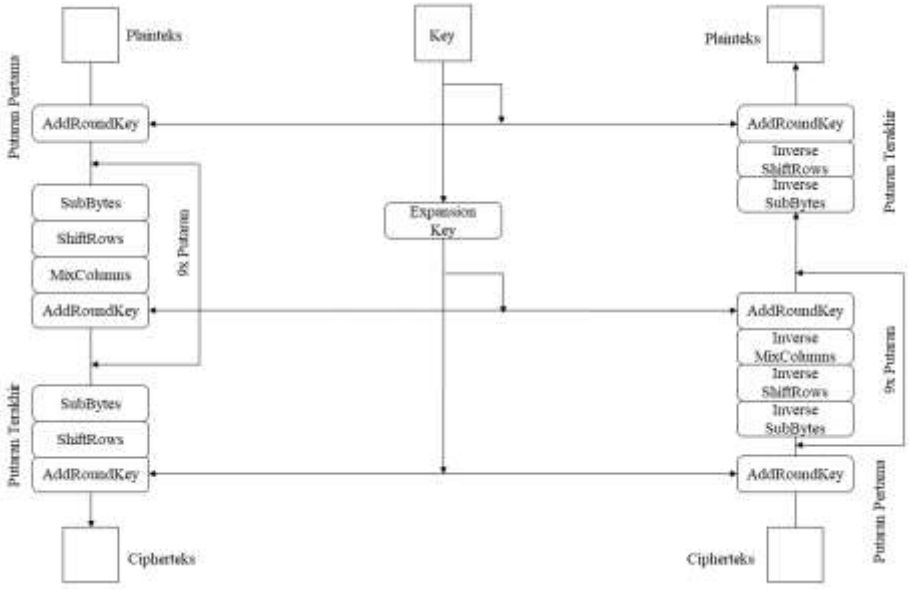
Proses *SubBytes* adalah operasi yang akan melakukan substitusi dengan cara mengganti setiap *byte state* dengan *byte* pada sebuah tabel yang dinamakan tabel S-*Box*. Sebuah tabel *S-Box* terdiri dari 16x16 baris dan kolom dengan masing-masing berukuran 1 *byte*.

3. *Shift Rows* (Transformasi Pergeseran Baris)

   Proses *ShiftRows* akan beroperasi pada tiap baris dari tabel *state*. Proses ini akan bekerja dengan cara memutar *byte* pada 3 baris terakhir (baris 1, 2, dan 3) dengan jumlah perputaran yang berbeda-beda. Baris 1 akan diputar sebanyak 1 kali, baris 2 akan diputar sebanyak 2 kali, dan baris 3 akan diputar sebanyak 3 kali, sedangkan baris 0 tidak akan diputar.
4. *Mix Column* (Transformasi Pencampuran Kolom)

   Proses *Mix Columns* akan beroperasi pada tiap kolom dari tabel *state*. Proses ini mengoperasikan 4 *bytes* dari setiap kolom tabel state dengan matriks *Mix Columns* dalam AES. Kecuali tahap *Mix Columns*, ketiga tahap lainnya akan diulang pada setiap proses, sedangkan tahap *Mix Columns* tidak akan dilakukan pada tahap terakhir.

Proses dekripsi adalah kebalikan dari enkripsi. Transformasi dibalikkan dan diimplementasikan kearah yang berlawan untuk memperoleh *inverse cipher* [9]. Proses enkripsi dan dekripsi pada AES lebih lanjut dapat dilihat pada Gambar 1.



Gambar 1. ***Proses Enkripsi dan Dekripsi pada AES***

Untuk melakukan pengembangan jumlah kunci yang akan dipakai dari kunci utama maka dilakukan *key schedule*. Proses ini terdiri dari beberapa operasi, yaitu:

1. Operasi *Rot Word*, yaitu operasi perputaran setiap bytes dari kunci secara siklik.

$$RotWord(B_0, B_1, B_2, B_3) = (B_1, B_2, B_3, B_0) \quad (8)$$

2. Operasi *SubBytes*, pada operasi ini 8 bit dari *subkey* disubstitusikan dengan nilai yang ada pada tabel dari S-*Box*.

$$SubWord(B_0, B_1, B_2, B_3) = (B_0', B_1', B_2', B_3') \quad (9)$$

   dengan $B_i' = SubBytes(B_i), i = 0, 1, 2, 3.$
3. Operasi XOR dengan w[i-Nk] yaitu word yang berada pada Nk sebelumnya.

4. Operasi *Rcon*, merupakan operasi XOR dengan nilai pada Tabel *Rcon* sesuai dengan *round* yang dijalankan. Tabel *Rcon* dapat dilihat pada Tabel 2. Nilai-nilai dari *Rcon* kemudian akan di XOR dengan hasil operasi sebelumnya.

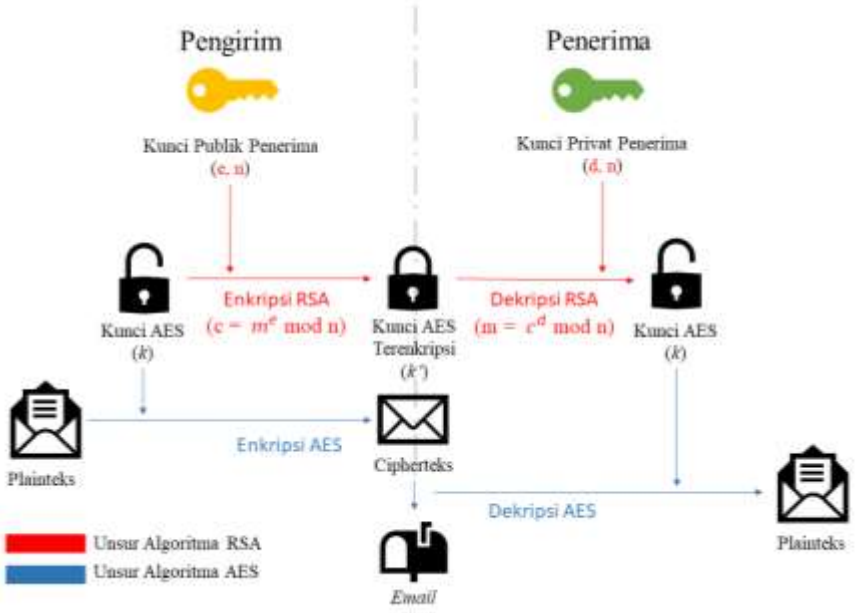
Tabel 2. **Nilai Rcon**

Round	1	2	3	4	5	6	7	8	9	10
Rcon[]	01	02	04	08	10	20	40	80	1b	36
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00

AES juga menerapkan operasi *bitwise*, yaitu operasi pada level bit (1 dan 0) dalam representasi biner dari data, memungkinkan manipulasi langsung terhadap bit-bit tersebut. Operasi *bitwise* memungkinkan implementasi perangkat keras khusus yang dioptimalkan untuk AES, karena perangkat keras dapat didesain untuk melakukan operasi *bitwise* secara paralel dengan kecepatan tinggi. Salah satu kekhawatiran yang muncul adalah ketika kunci rahasia AES harus ditransmisikan dari satu pihak ke pihak lain secara aman.

HASIL DAN PEMBAHASAN

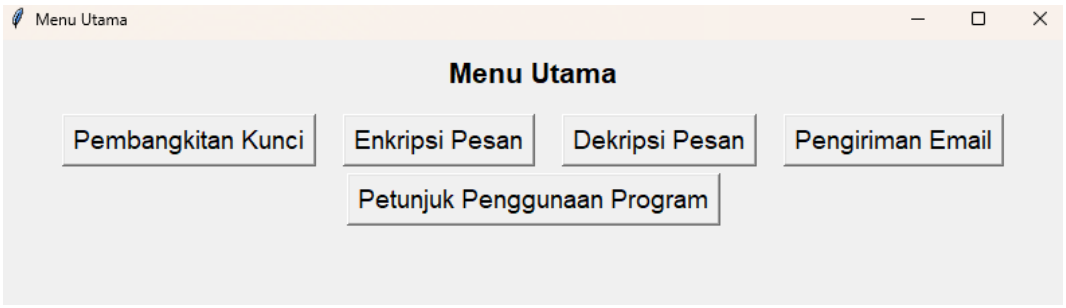
Implementasi penggabungan algoritma kriptografi RSA yang ditingkatkan dan AES bertujuan untuk meningkatkan keamanan pada pesan rahasia. Dari permasalahan transmisi kunci yang ditemukan saat implementasi algoritma AES, maka algoritma kriptografi RSA akan diterapkan agar pertukaran kunci AES lebih aman. Selain itu, karena algoritma kriptografi RSA memiliki tahap-tahap komputasi yang berat, algoritma ini akan lebih cocok diterapkan pada kunci karena kunci yang akan digunakan berukuran tetap yaitu 16 *bytes* (AES-128). Penggabungan kedua algoritma kriptografi ini disebut juga dengan kriptografi *hybrid*. Cara implementasi kriptografi *hybrid* ini ditunjukkan dalam skema pada Gambar 2.



Gambar 2. Skema Penggabungan Algoritma RSA yang Ditingkatkan dan AES

Pembuatan program aplikasi dari algoritma kombinasi hasil penggabungan algoritma RSA yang ditingkatkan dengan algoritma AES dilakukan dengan cara mengubah model matematis algoritma kombinasi ke dalam bahasa pemrograman *python*. Program dikonstruksi menggunakan *Python* Versi 3.9.6 pada komputer dengan sistem operasi *Windows* 11 64-*bit, processor* Intel Core i5-11357G dan RAM 8 GB.

Program yang telah dikonstruksi memiliki beberapa pilihan menu mulai dari pembangkitan kunci, enkripsi, dekripsi hingga pengiriman pesan melalui *e-mail*. Tampilan awal dari program ditunjukkan pada Gambar 3.



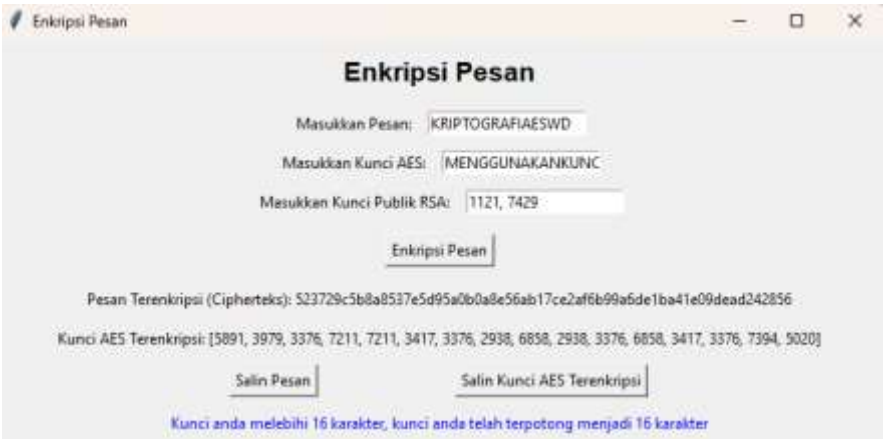
Gambar 3. Tampilan Menu Utama

Pada program pembangkitan kunci, penerima hanya perlu memasukkan tiga bilangan prima yang dipilih lalu diperoleh kunci publik dan kunci *privat*. Kunci publik yang telah diperoleh kemudian dikirimkan ke pengirim pesan, sedangkan kunci *privat* dirahasiakan dan disimpan oleh penerima pesan. Pada Gambar 4 ditunjukkan contoh penggunaan program pembangkitan kunci dengan menginput bilangan prima $p = 17, q = 19, r = 23$, diperoleh Kunci Publik (1121, 7429) dan Kunci *Privat* (3233, 7429).

Gambar 4. Tampilan Menu Pembangkitan Kunci

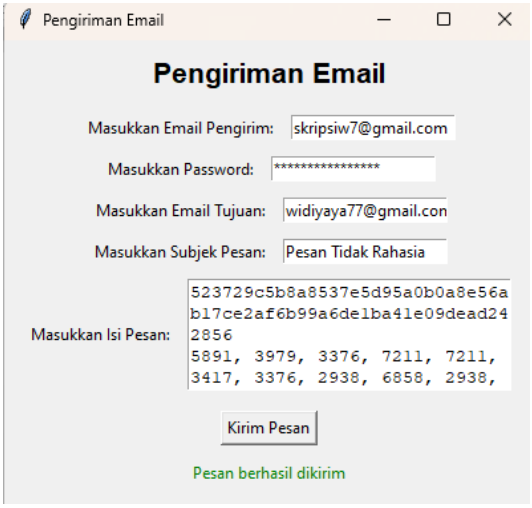
Pada program enkripsi pesan, pengirim pesan menginput pesan yang akan dikirimkan, kunci AES yang dipilih oleh pengirim, dan kunci publik yang diperoleh dari penerima pesan, kemudian akan diperoleh kunci AES terenkripsi dan *cipherteks*. Ditunjukkan contoh *input plainteks* "KRIPTOGRAFIAESWD" dan kunci AES "MENGGUNAKANKUNCII", dan kunci publik dari penerima pesan "1121, 7429". Kunci AES terpotong otomatis menjadi 16 karakter, lalu diperoleh pesan terenkripsi "523729c5b8a8537e5d95a0b0a8e56ab17ce2af6b99a6de1ba41e09dead242856" dan

kunci AES terenkripsi "5891, 3979, 3376, 7211, 7211, 3417, 3376, 2938, 6858, 2938, 3376, 6858, 3417, 3376, 7394, 5020" pada Gambar 5.



Gambar 5. Tampilan Enkripsi Pesan

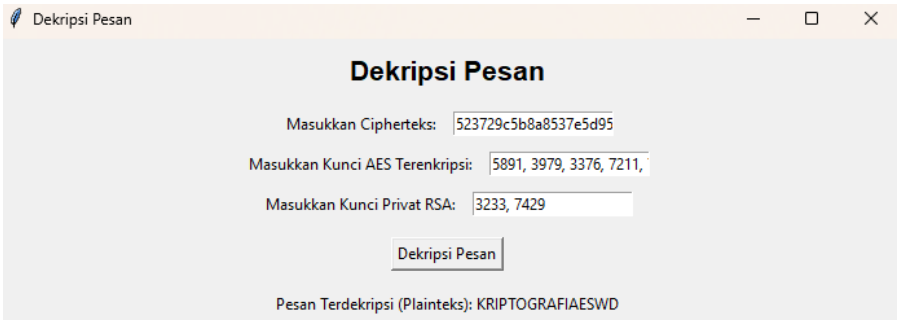
Pada menu pengiriman *e-mail*, pengirim pesan menginput e-mail pengirim, *password e-mail*, *e-mail* tujuan, subjek pesan, dan isi pesan. Isi pesan terdiri dari *cipherteks* dan kunci AES yang terenkripsi. Ditunjukkan contoh input skripsiw7@gmail.com sebagai *e-mail* pengirim, diinput *app password*, kemudian *e-mail* tujuan widiyaya77@gmail.com, subjek pesan dicontohkan "Pesan Tidak Rahasia", dan isi pesan "523729c5b8a8537e5d95a0b0a8e56ab17ce2af6b99a6de1ba41e09dead242856, 5891, 3979, 3376, 7211, 7211, 3417, 3376, 2938, 6858, 2938, 3376, 6858, 3417, 3376, 7394, 5020" pada Gambar 6. Tombol "Kirim Pesan" ditekan untuk mengirim pesan. Setelah *e-mail* terkirim, pengguna akan muncul teks "Pesan Berhasil Dikirim".



Gambar 6. Tampilan Pengiriman *E-mail*

Pada menu dekripsi pesan, penerima pesan menginput *cipherteks* dan kunci AES terenkripsi dari pengirim pesan, dan menginput kunci *privat* yang sebelumnya sudah dimiliki. Dicontohkan pada Gambar 7 *input cipherteks*, dan kunci AES terenkripsi adalah "523729c5b8a8537e5d95a0b0a8e56ab17ce2af6b99a6de1ba41e09dead242856", dan "5891, 3979, 3376, 7211, 7211, 3417, 3376, 2938, 6858, 2938, 3376, 6858, 3417, 3376, 7394, 5020", kemudian "3233, 7429" sebagai kunci *privat*. Diperoleh *plainteks*

bertuliskan “KRIPTOGRAFIAESWD”.



Gambar 7. Tampilan Dekripsi Pesan

KESIMPULAN

Berdasarkan penelitian yang dilakukan diperoleh kesimpulan bahwa implementasi penggabungan kriptografi RSA yang ditingkatkan dan AES pada aplikasi pengirim *e-mail* dilakukan dengan mengkonstruksi suatu program aplikasi komputer dengan bahasa pemrograman *Python*. Dalam aplikasi tersebut, pengguna dapat melakukan pembangkitan kunci, enkripsi pesan, dekripsi pesan, dan pengiriman *e-mail*.

Dalam penerapannya, skema penggabungan kriptografi RSA yang ditingkatkan dan AES dimulai dengan adanya perubahan pada pembangkitan kunci, di mana pada algoritma RSA standar digunakan dua bilangan prima, sedangkan pada algoritma kriptografi RSA yang ditingkatkan digunakan tiga bilangan prima. Hal ini dapat membuat proses faktorisasi lebih sulit karena harus memfaktorisasi menjadi tiga bilangan. Setelah pembangkitan kunci, skema kriptografi *hybrid* diterapkan pada penggabungan dua algoritma ini di mana kunci publik RSA yang ditingkatkan akan digunakan untuk enkripsi kunci AES sehingga permasalahan transmisi kunci rahasia dalam algoritma AES dapat terselesaikan.

DAFTAR PUSTAKA

[1] Wahyadyatmika, A. P., Isnanto, R. R., & Somantri, M. (2014). Implementasi Algoritma Kriptografi RSA pada Surat Elektronik (E-mail). *Transient: Jurnal Ilmiah Teknik Elektro, 3*(4), 442-450.

[2] Firdaus, J., Marwati, R., & Gozali, S. M. (2018). Penyandian Pesan Menggunakan Kombinasi Algoritma RSA yang Ditingkatkan dan Algoritma ElGamal. *Jurnal EurekaMatika, 6*(1), 23-32.

[3] Fatma, Y., Hafid, A., & Dani, H. O. (2020). Peningkatan Keamanan Pengiriman Pesan Teks: Kombinasi Advanced Encryption Standard (AES) 128 dan Least Significant Bit (LSB). *JUSIFO (Jurnal Sistem Informasi), 6*(2), 111-120.

[4] Bimantoro, Y., & Sari, R. T. (2021). Enkripsi Data Menggunakan RSA & AES Pada Aplikasi Instant Messaging Berbasis Mobile. *Jurnal Teknik Informatika, 14*(2), 135-144.

[5] Hermawan, A., & Ujianto, E. I. (2021). Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan, 5*(2).

[6] Basri. (2016). Kriptografi Simetris dan Asimetris Dalam Perspektif Keamanan Data dan Kompleksitas Komputasi. *Jurnal Ilmiah Ilmu Komputer, 2*(2), 16-23.

[7] Surian, D. (2006). Algoritma Kriptografi AES Rijndael. Tesla: Jurnal Teknik Elektro, 8(2), 97-101.

[8] Anwar, N., Munawwar, Abduh, M., & Santosa, N. B. (2018). Komparatif Performance Model Keamanan Menggunakan Metode Algoritma AES 256 bit dan RSA. *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi), 2*(3), 783-791.

[9] Hutasuhut, D. I., Aldizar, M. R., & Nasution, I. F. (2023). Perbandingan Algoritma Kriptografi Simetris dan Asimetris. *UNES Journal of Information System*, *8*(1), 042-047.

[10] Nasution, R., & Triandi, B. (2020). Implementasi Metode RSA dan AES untuk Mengamankan File Winrar dan ZIP. *IT (Informatic Technique) Journal*, 2252-746X.